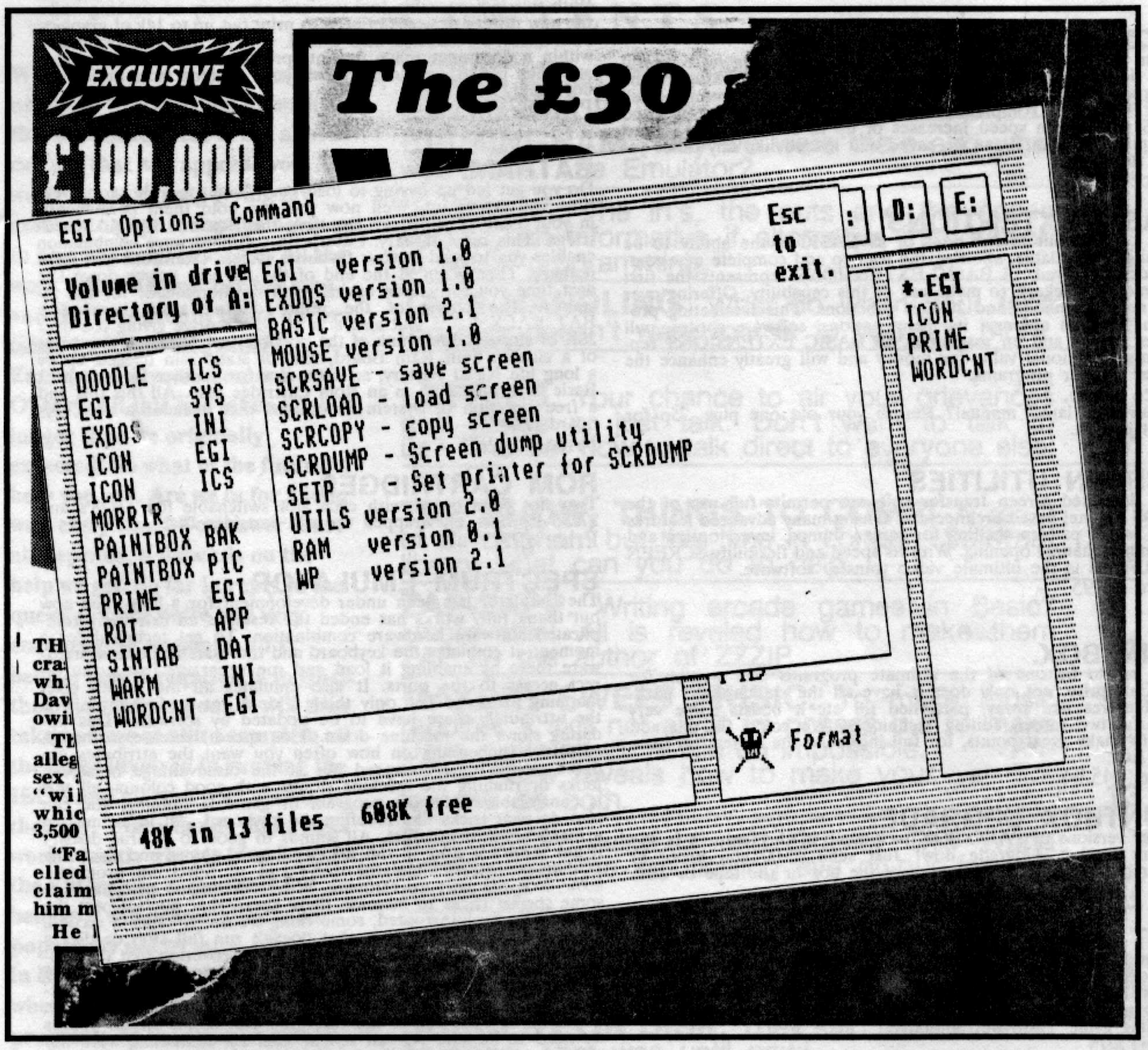
No. 1 February 1989

Enterprise Computing International



```
EGI  Options  Command

Volume in drive
Directory of A:

DOODLE    ICS
EGI       SYS
EXDOS     INI
ICON      EGI
ICON      ICS
MORRIR
PAINTBOX  BAK
PAINTBOX  PIC
PRIME     EGI
ROT       APP
SINTAB    DAT
WARM      INI
WORDCNT   EGI

48K in 13 files    608K free

EGI     version 1.0
EXDOS version 1.0
BASIC version 2.1
MOUSE version 1.0
SCRSAVE  - save screen
SCRLOAD  - load screen
SCRCOPY  - copy screen
SCRDUMP  - Screen dump utility
SETP     - Set printer for SCRDUMP
UTILS version 2.0
RAM    version 0.1
WP     version 2.1

Esc
to
exit.

:   D:   E:

*.EGI
ICON
PRIME
WORDCNT

Format
```

New graphic environment. Scoop.

Introduction to IS-BASIC, tutorial starts here.

Zappp! How to write games in BASIC by Peter Hinner, the author of Zzzip.

Why write your own music when you can get DAVE to do it for you?

This issue we feature cassette care and tape to disk conversion.

BoxSoft
PRESENTS

ZZZIP

Open up your Enterprise by dramatically speeding its powerful and flexible IS BASIC language. Simply load ZZZip before the program you want compiled and this comprehensive compiler will over 90% of all comprehensive of all functions such as SIN and COS resulting in speed increases of up to 5000%. Once loaded compiled IS BASIC can be saved and loaded like any other yet will run faster.

Price £12.95.

BASIC EXTENTIONS

One of the built in functions of IS BASIC is the ability to be infinitely extendable, allowing the user to add complete new commands as desired. IS BASIC EXTENSIONS represents the first commercial package to make use of this capability. Offering over 100 new commands and 20 new functions. This fascinating program give you amongst its many options; software sprites, pull down menu's and an assembler. IS BASIC EXTENSIONS represents exceptional value for money and will greatly enhance the power of your programs.

Price £7.95.

Require the latest manual? Return your old one plus .75p for latest edition.

SCREEN UTILITIES

This dedicated screen transfer software permits full use of the EXOS channel based architecture. Giving many advanced features such as 256 pattern shading for screen dumps, invert option and automatic channel opening. With its speed and flexibility SCREEN UTILITIES is the ultimate video transfer software.

Price £5.95

PAINTBOX.

This has to be one of the ultimate programs ever written for the Enterprise, not only does it have all the standard art package features like spray, patterned fill etc it boasts some very powerful two screen editing techniques. We could fill this add with its make great points, for full info read the review in issue 1 of ECI.

Price £45.94

PAINTBOX UPGRADE.

Earlier versions of PAINTBOX lack some of the features in the current version. Upgrade now! Just send £3.00 and proof of ownership, say the outside cover of the box or the tape or disk.

OUT OF THIS WORLD.

Far away in a galaxy many light years away from our own, war has torn apart many worlds. Stars lie desolate and lifeless. It seems that nocreature or machine has survived!!!! The story may be no good but the game is brilliant. Ball Blazer style bouncing Scrolling game. Hard but addictive.

Price 7.95

TECHNICAL MANUAL.

Discover all about the Enterprise and Exos with the Enterprise technical manual. Note. Only recommended for the technically minded

PRICE £9.95

At long last we can bring you four new games from our friends in Hungary...

MIRROR WORLD.

Nasty aliens are invading the galaxy, you have to shoot them down, the only trouble is your lazar isn't powerful enough so bounce them of the sky mirrors to boost there energy.

EGGS OF DEATH.

Multi level game. Starting with a version of Glaxions, moving on to a Platform style game and ending up with Scrambler.

ENTERBALL.

Enterprise version of the classic. 'BREAKOUT' Many levels, great fun to play.

NATILUAS.

Take control of your own Submarine, Hunt out the baddies and blast them out of the water.

ALL PRICED AT £5.95

PRINTER UTILITIES.

Are you always waiting for your printer? Do your documents look drab and boring? Then we prescribe PRINTER UTILITIES. With this indispensable tool you can now carry on working whilst the new printer driver continues to print out up to 14k of graphics. PRINTER UTILITIES also allows you to place control codes within a document ready for interpretation whilst being printed and thus giving you access to all your options from within the document.

PRICE £7.95

BATRAM.

Do you get fed up having to load your favourite utilities in every time you power up. Well now you can hold them permanently in your Enterprise. How? By using the BoxSoft BATRAM cartridge. This revolutionary, Enterprise soft/hard ware combination enables you to load in your favourite system extensions and leave it there. Then when at the end of the day you power down.The next time you turn it on again it will be there ready for use instantly. The power of the utility doesn't stop there! The BAT BACK RAM unit also acts as a Ram drive giving you over 28K of storage. Just think of the possibilities! The unit comprises of a custom built Ram Board, 32K of static ram backed up by a long life Nicad Battery, an extra slot for another Rom such as Basic all wrapped up in an Rom Cartridge case. All this this and a free selection of System Extensions that includes a new keyboard Driver.

Price £24.95

ROM CARTRIDGES.

Two slot Rom board with each slot switchable for 27128's and 27256 Eproms, all wrapped up in a cartridge case.

Price £7.95

SPECTRUM EMULATOR.

The Emulator has been under development for a long time now but thank fully works has ended the result is an extremely complicated software hardware combination. To get technical for a moment, it emulates the keyboard and the cassette ports in hardware there by enabling it load and run programs that make direct access to the ports. It also emulates all the screen pixel mapping properly. The only thing it does not do in hardware is the attributes, these have to be updated by software. This updating slows the machine down. This speed decrease is though adjustable depending on how often you want the attributes updated slowly for high speed but at the disadvantage of screen looks or running the machine slower with good colours, all this is controlable while the program or game is running, you can also do neat tricks like redefine the keyboard. All in all making a really powerful product. All sounds to good to be true doesn't it but there's a sting in the tail of all good things and this sting is that the speccy emulator won't run all Spectrum programs, 80% + is the best we estimate. Why? Well some programs use some special tricks to increase there speed that don't expect the program to be interrupted, some thing we do 50 times a second, the net result is a program that doesn't run full stop or dies a slow agonising death. To prevent people discovering that the program doesn't work after they've spent a tenner on it, we will be producing a software compatibility list. This list will be constantly up dated and available on request free.The last thing is when it will be on sale, due to the very high cost of the circuit boards we need to be sure we can sell at least 20, something were unsure of at present, so if your intersected write to us here at BOXSOFT with a preliminary order ie just to say your are thinking about purchasing one. If you are in doubt look out for previews on the prototypes, to be published in forth coming issues of ECI.

Price about £45.00

PRINTER CABLE.

The Enterprise can be connected to any printer via one of its many PORTS. The Enterprise printer cable will enable you to connect your Enterprise to all those with a CENTRONICS connector

Price £9.95

All prices include P+P.

BOXSOFT, 60 HOLDENHURST AVE,

LONDON N12 OHX

Enclose order, address and cheque or postal-order.

No. 1 February 1989

Enterprise Computing International

CONTENTS

Well we made it at last. I know by now your probably fed up with all this delay and there is not a lot I can say that will appease you. So I won't spend to long giving my excuses, except to say the delay up to now has been caused by all the work (putting together the mag and the user group) being under taken by myself and other Enterprise devotee Eamonn O'leary. All this has taken a lot longer than we originally expected! So what of the future? I here you ask. Are we in for a long wait every time? To answer that all depends, it depends on the help we get. So far in reply to the questionnaire we've had very encouraging response, all I believe now is required is to lynch them into helping out, it will then take no time at all to complete all the tasks involved in running the IEUG and all that goes with it, as they say, "many hands make light work". Lastly what is in store for the Enterprise in general? I believe it will see an up turn in popularity not by more being sold in England but by people every where, seeing through such great programs as Paintbox and EGI that the Enterprise is one hell of a machine and is not going to lie down and die just because its not on computer shops shelves.

ECI is a IEUG publication. Production team:- Magazine Editor, Publisher, Tim Box. Artwork and layout Eamon O'Leary, Tim Box. Contributors (in alphabetical order) Tim Box, Andy Burnham, Peter Hinner, Anthony Shaw, David Silkstone. Rei Wilkenson. No article may reproduced in whole or part without written consent from the copyright holders.

The Independent Enterprise Users Group currently resides at:

60 Holdenhurst ave,
Finchley, London, N12 OHX, ENGLAND.

Any questions pertaining to Enterprises or the IEUG should be directed to Tim Box either by writing to the above address or for a quicker reply by telephoning 01-346-8032 between the hours of 6pm to 9pm Mon to Friday. Please note any phone calls out side those hours will not be received happily.

NEWS

Hungarians break down the wall.

Good news on the software front as Hungarian programs finally reach English shores. BoxSoft, via its Denmark subsidiary BoxSoft DK, is selling Hungarian programs from the software company the "A" Studio.
The four programs are Nautilus; a submarine simulation, Enter Ball; a break out style arcade game, Mirror World and Eggs of Death; both arcade style "shoot'em ups". The games are all priced at £5.95. Watch out for full reviews in the next news letter.

"Enterprises" Where are they now?

Did you ever wonder what became of all the Enterprises unsold when Enterprise Computers went into liquidation?
The answer is Hungary! Just as the receivers were marching to the front door a lot of computers were on their way out, through the back door. This small number of machines together with a larger number purchased from the liquidators (via the front door) have all found there way over the Viena border into Hungary. The Enterprise is now, as far as I am aware, one of only two official micros in the country. You may well be asking what the first is? Don't. Why? Well, to be honest, I've forgotten it's name. I can tell you one thing though it has very close ties with our own beloved micro. The connection? Intelligent Software, Designers of both!

Emulator finished?

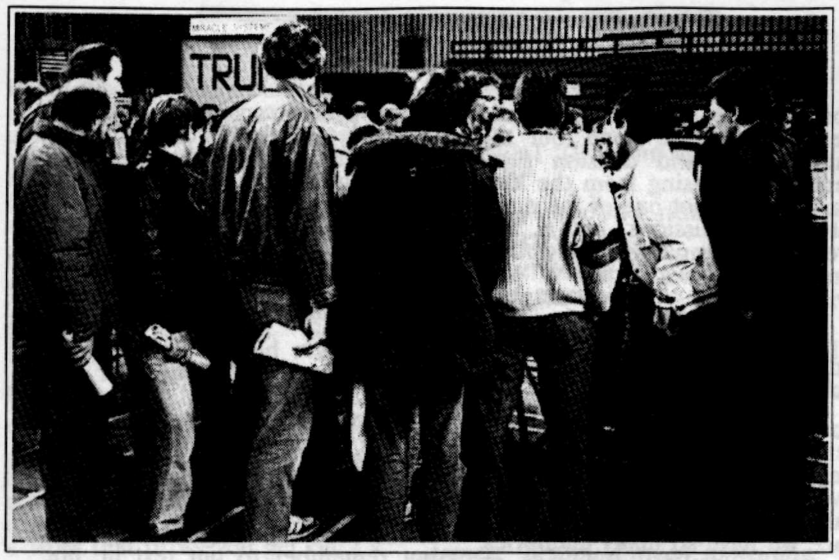
As mentioned in other places within this publication, the BoxSoft Spectrum Emulator is finished.
How finished? Well, according to Tim Box all that is required to do is to manufacture it. This may not sound like it's finished. To this point he said "To make them economically viable ie, less than 50 each the circuit boards have to made in quantities of 20 or more. So before the readies are given out to the PCB manufacturer he has to be sure he will sell them. The intervening time will however be put to good use in compatibility testing which by the way stands at about 80%". When questioned more about compatibility, he hesitated. We have discovered that some programs use very complicated techniques to increase the speed of programs, this has however caused problems. In the course of the Emulators taking control of the Z80 4 bytes of memory are over written, while in 80% of the programs this makes no difference to the running of the program, in the remaining 20% it causes the corruption of the games code and data. There is no way of getting round this. To prevent money being wasted on games that don't work we are compiling a list of all the games tested and shown to work, this list will be added to constantly and be made available to everyone upon request. For more information about the Emulator Tim Box can be contacted on 01-346-8032.

EGI - cover story.

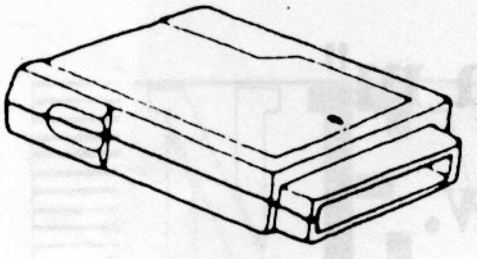
As you will have seen from the front cover of this Publication the most exciting new product soon to be released is EGI (short for Enterprise Gaphical interface).
We cannot compressively describe the program here because it is undergoing a major rewrite, but to give you a flavour of things to come this is a run down of how it will work.

EGI is as the title implies, a graphical front end. In use programs using EGI just make a series of system calls specifying what it wants done. This will range from adding new icons to its icon table to displaying them and monitoring the mouse movements to providing pull down menus. From this basic interface, programs can be built up with a very professional look and feel to them; it also means you don't have to re-invent the wheel every time you want to provide a fancy display. There has already been several programs written to run under EGI, they include a File Manager an Icon Editor and a full A4 page size screen designer, with multiple fonts and the ability to import pictures. The quality of this program has to be seen to be believed. As mentioned above it is not yet on sale but should be soon. We will keep you informed of it's progress and will of cause be giving it a full review in the a forth coming magazine.

"Birmingham" The show.



Saturday the 12th of November started for me at the ungodly hour of 5.30am, this was the time I had to get up to enable the transportation of my self, Eamonn O'Leary and half a car load of Enterprise bits and pieces up the M6 to the Aston Villa Sports and leisure Center in Birmingham. For those of you who did not receive a news letter with this venue mentioned on it, it was of course the setting for the Alternative Micro Show. The show, as the title implies, was for the likes of us Enterprise users; micro owners whose manufacturer no longer exists or no longer supports there machine. The show organised by Emsoft attracted many micros clubs supporting lots of machines from the past. Upon arriving we discovered that far from being a small gathering with small groups of people like ourselves, there were large companies unloading boxes and boxes of computers and computer related goods. This initial impression of a very professional gathering was then quashed when we looked at our stand. One small table about six foot long that looked like it been venue for a lumberjacks tap dancing contest (there were that many foot prints on it). Still I must not grumble as it did give us an opportunity to have a show and meet up with people again. Going back to the stand again, we then proceeded to assemble something of a display. Those of you who went will know exactly what it ended up like, those of you who did not will see what we achieved from the accompanying photos! Just in the nick of time before the show was about to open two knights in shining armour arrived in the guise of Peter Hiner (author of ZZZIP) and Andrew Richards (author of Out of This World among other things). With their help we managed to assemble all the equipment together in a form that eventually worked if you ignore the odd power cut. From 10am when the doors opened the stand as you will see from the pics was awash with users. Before I continue discusing what was on display at the show I would like to thank all those of you who turned up, it was great to see all of you and if there were any doubts to whether it was worth while taking the stand they were very quickly quashed by the enthusiasm of all those who attended. There were two major products displayed at the show. One was the long (I should say very long) awaited Spectrum Emulator, displayed in two forms; one prototype working and one not-so-prototype not so working! We managed to demonstrate it loading programs with hyper-loaders and loading Speccy programs from disk (not a feat to be scoffed at). The other star of the show was EGI (short for Enterprise Graphical Interface). EGI, as the name implies, is a graphical front end for the Enterprise in the style of GEM, WINDOWS etc. It really is a great piece of work, the only disadvantage is that like the programs it emulates it is disk based and at the present time requires a Mouse. If there was any reason to justify the purchase of an EXDOS and Mouse this has to be it. On the subject of EGI there should be more news of it elsewhere in this publication. There were a small number of programs on sale at the show but alas they were old ones, for example ZZZip, OTW, (reviewed in this issue) Screen utilities and couple of others that have escaped my memory. The demand for even these old titles did bring one thing to my attention. More new programs need to be published as there is obviously a demand. One of the products that did generate a lot more interest than originally anticipated was Exdos the disk interface. As you will see from the BoxSoft advert I am going some way to solving the problem of diminished supplies. Looking round at the other exhibits (when I had time) it became very apparent that whilst other micros like the Einstein (which managed to sell in very large numbers in comparison to the Enterprise) were only selling and showing old programs and peripherals the Enterprise was making its mark as a machine that's still having new programs developed for it. Lastly I would like to thank again all those who made the effort to attend and support us showing (to me at least) that the Enterprise is still alive and kicking.



PART ONE

An Introduction to IS-BASIC

By Andrew Burnham.

Once upon a time there was a little computer called the Enterprise. It was a very hippy computer. All the most drug crazed beardy wierdies in an economic wasteland called Thatcherist Britain had worked for at least five minutes, including coffee breaks to make the Enterprise the best computer ever.

However the second smuggled shipment of high grade caffeine from Hong Kong was intercepted by customs and the Enterprise was launched on an unsuspecting world in its underwear, but still better than the average home computer (some being more average than others). Today you're learning to write BASIC on the Enterprise, and are possibly on the first step of the long downhill struggle to beardy wierdydom. Aren't you lucky?" (Basic manual p77)

Right. You still want to be a programmer? How many coffee cup rings are there on the top of your Enterprise? None? Well go and get yourself a mug of very strong coffee. NOW! All programmers consume huge quantities of coffee, probably to help them keep awake on all-night programming sessions. If you don't like coffee then try intravenous caffeine, or stop trying to be a programmer!

Firstly, although seeming backwards, we'll get BASIC to tell us things. It may seem odd as it hasn't yet got anything to tell us, but bear with me for a while. I'm sure it was long enough before you said anything other then 'Dada'.

The command to make BASIC say anything is 'PRINT'. Try entering following line :

```
PRINT "Dada"
```

Now, doesn't that make you feel proud? If that didn't work then try it again, repeated failure probably means that you haven't got BASIC loaded, the machine turned on, or you are suffering from a severe case of brain death.

PRINT will do other things than just repeat you. If you are really pushed you can use it as a very expensive calculator. Try this :

```
PRINT 3+4
```

Most of the usual mathematic operations can be done, but the multiplication and division signs seem to be missing from the keyboard. This is not one of the many Enterprise omissions, but a 'feature' of all computers. Multiplication is done with the '*' and division with the '/'. As an example try subtracting the cost of an Enterprise from the cost of a BBC, and see how much pocket money you've saved yourself.

More advanced calculators have a memory to store a number for later. Very expensive ones may have several. BASIC has one less than more than you will ever need. Since we have so many memories, we must give them names to remember which is which. A memory or 'variable' name can be one or more words and numbers, but it must start with a letter and not contain any spaces. Spaces can be replaced with the '_' character. Some examples of names are :

```
Crunchy_frog
Tel_0101_516_976_2828
Pooey_port_3 X
```

unfortunately names are limited to a maximum of 31 characters, so a name like

```
Lamborgucci_midnight_blue_
multiform_featherweight_
jumbo_ribtickler
```

would be too long.

To put a number in a variable we use a 'LET' command like this:

```
LET Editors_mental_age = 4
```

In this command the '=' is NOT the mathematical symbol. The two sides of the '=' are not necessarily equal. Think of it as 'becomes' rather then 'equals'. Note that the word LET may be omitted by bad programmers, which will cause confusion.

To use the contents of a variable we just put it in place of the number required, e.g.

```
PRINT Editors_mental_age
```

This should print '4', just like we set it earlier. If you think this a bit generous then we can downgrade it a bit with:

```
LET Editors_mental_age =
Editors_mental_age - 1
```

If you print it again you will find that it is a little bit closer to the truth. This is where the equals sign is obviously not as in mathematics, since something cannot equal itself minus one.

Right, that's enough maths for now, besides my coffee's gone cold with the mental strain of it all.

We can save bits of text for later as well, so shove that up your calculator and smoke it! The only difference is that the name of a 'string' variable has a '$' sign at the end of it. This helps both you and BASIC to remember that it's not a number. We use a LET statement for storing these as well, but remember to put text in, not a number otherwise BASIC will scream at you.

```
LET A$ = "Power"
LET B$ = "house"
```

This gives us two 'strings' which we can work with. To print one we just type:

```
PRINT A$
```

Which will print 'Power'. We can add stings together, with '&', which simply tacks the second string on after the first, e.g:

```
LET Neils_company$ = A$ &
B$
```

PRINTing Neils_company$ will give you the answer, which you should already recognise. We can chop bits out of strings as well, Marvel Comics got good mileage out of the following:

```
PRINT A$ (1:3)
```

Which gives letters 1 to 3 of the string A$. Several things can be done at once to give where Neil lives:

```
PRINT A$ ( :2) & B$
```

Note that I've only given the last character to print; if you don't give the first, the 'default' is the first character of the string. The default for the last character is.. ...the last character of the string of course.

At this point we will have a short break since all this coffee has started to get to me. Practice what you've learned, I'll be right back after this word.................. .

£ £ £ £ £ £ £ £ £ £ £

£ £ £ £ £ £ £ £ £ £ £

Sooper soundz, yes indeedy, greatch to be with you after the break. Now we can get to a program. Yes, I know that I've spelled it wrong but that's to distinguish it from that dross you get on TV.

A program is simply a set of commands stored to do later. This is what really makes a computer better than a calculator. Once you have written a program to do something, you will never have to do it again (barring a system crash, or coffee spillage).

To tell the computer to save an instruction we just put a number before it:

```
10 PRINT "Hello"
```

This time nothing seems to have happened. However BASIC has remembered the line for use later. To see the stored line type 'LIST', which will print the line you have just typed. To actually do the command enter 'RUN', the word 'hello' should be printed.

Now that's what I call a 'user friendly' program. The following is an example of a 'user hostile' program:

```
10 PRINT "Clear off!"
```

Of course, one line is not really all that useful. That's where the line number comes in. Stored lines are sorted into order of line numbers, and when you RUN the program, BASIC works through from the first to the last in order, unless you tell it otherwise, which I shall come to later.

There are some more commands which are only really useful in a program. The most interesting one for now is 'INPUT', which makes the program ask for numbers or strings. Try putting in the following program, after entering 'NEW' to delete the old one:

```
10 PROGRAM "Simple sums"
20 PRINT "Give me a number"
30 INPUT first_number
40 PRINT "Give me another
number"
50 INPUT second_number
60 PRINT "Sum is ";
first_number +
second_number
70 PRINT "Difference is ";
first_number - second_number
80 END
```

Now, a full explanation of the program, line by line.

10 - *A name should be given to all BASIC programs for further reference. It should mean something, and not be obscure like "Fred".*

20 - *An instruction to the 'user'*

30 - *Get a number from the user, and put it in a variable called 'first_number'.*

40 - *Another instruction*

50 - *Another input*

60 - *Print some information. Note that you can print two things at once by separating them with a ';'.*

70 - *More information*

80 - *A very important END statement. This makes BASIC stop running the program. A lot of bad programmers miss it off. BASIC should stop when it falls off the end of the program, but you can't complain at it if it doesn't.*

Try running the program a few times, giving it different numbers to see what it's doing. LIST it as well for good measure.

The INPUT statement can display its own instructions instead of using a separate PRINT. To save ourselves a line we can delete line 20 by entering just 20 on its own, and replace line 30 with:

```
30 INPUT PROMPT "Give me a
number " : first_number
```

You can do this with lines 40 and 50 as well. This means that the line numbers are uneven. BASIC won't mind as it doesn't have a sense of aesthetic proportion, and judging from a lot of BASIC programs, neither have many programmers. To tidy up we can enter RENUMBER which will space them evenly again.

Run the new program to check that it does exactly the same as before.

Well that was interesting wasn't it? That is a piece of 'linear' program, where all the instructions follow from one another in sequence. A further help is to be able to repeat a section of code several times to do a repetitive task. For example we can write the following program to help us test for palindromes (ie. words which are the same spelt backwards):

```
10 PROGRAM "Palindromes"
20 INPUT PROMPT "Give me a
word :" : word$
30 ! Now to reverse the word
(one character at a time)
40 LET reverse$ = "" ! Create
a blank string
50 FOR letter= 1 TO
LEN(word$)
60  LET reverse$ =
word$(letter) + reverse$
70 NEXT letter
80 PRINT ! Just a blank line
90 PRINT "Original word is
";word$
100 PRINT "Reversed word is
";reverse$
110 END
```

The program explanation is:

10 - *Program title*

20 - *Get a word to reverse*

30 - *A quick comment (please insert as many of these as humanly possible to help understanding)*

40 - *Generate a blank string to put the reversed word into.*

50 - *Step through the word letter by letter.*

60 - *Build up 'reverse$', by adding each letter of 'word$' on to the front of 'reverse$*

70 - *Loop back round to just after the FOR statement 'letter' times*

80 - *Print a blank line*

90 - *Print the original word*

100 - *Print the word backwards*

110 - *Don't forget the END statement*

In this program we are using a FOR - NEXT loop, which is stepped round 'letter' times. We could have put

```
50 FOR letter = 1 TO 4
```

This would have made the program loop four times, executing line 60 four times. Each time around the variable 'letter' increases by one, from one to four. The program would then only be able to handle four letter words, but most readers of this magazine would probably not notice the difference.

There is a second way to make programs loop around, this is with the DO - LOOP command. Here is the same program rewritten using DO and LOOP.

```
10 PROGRAM "Palindromes 2"
20 INPUT PROMPT "Give me a word :" : word$
30 ! Now to reverse the word (one character at a time)
40 LET reverse$ = "" ! Create a blank string
50 LET letter = 1
60 DO
70  LET reverse$ = word$(letter) + reverse$
80  LET letter = letter + 1
90 LOOP UNTIL letter len(word$)
100 PRINT  ! Just a blank line
110 PRINT "Original word is ";word$
120 PRINT "Reversed word is ";reverse$
130 END
```

The main difference between the two sorts of loop is that whilst FOR - NEXT increases a variable as it goes, DO - LOOP does not. Because of this we use an extra variable, still called 'letter', which we increase by one ourselves each time.

The LOOP UNTIL part of the structure is followed by what is known in the trade as an 'expression'. If this expression is false then the program will loop back to just after the DO command; if it is true, then the program will stop looping, and just continue with the next command (in this case line 100). There are lots of ways to make up the expression, which can use mathematical operators like "=", "", "", "=", "". The one we have used makes the program loop until 'letter' is greater than the length of the string 'word$'.

We'll continue in the next issue, but the commands introduced here should give you plenty to chew on, and you ought to be able to start writing your own simple programs.

Troubleshooters Guide to Tape Handling

By Rei Wilkinson

Have you ever had the feeling that your tape deck is if fact an incarnation of the Devil ? Perhaps you've spent a few evenings bashing your head against a wall because you want to play your favorite game, but your tape deck isn't on speaking terms with your computer ? Well, we can now offer a little friendly advice on house training your tape deck.

Seriously though, at some time we all experience a little difficulty with a tape deck, and this article is aimed at providing you with cures for most of the more common problems you may come across.

Firstly, can you load pre-recorded programs and packages, but not material that you have saved yourself? If this is the case, check that the 'OUT' socket of the computer is connected to the 'MIC' socket of the tape deck (or if your deck is like mine, the 'SAVE' socket).

If this is connected properly, check the deck itself. Some tape decks have what is called an A.L.C. recording system. This is a system that automatically controls the recording level (Auto Level Control). If your deck has this system, you will find somewhere on the deck itself a label saying A.L.C., or Auto Level Control. Decks that use this system are not usually the source of saving problems, although if you feel sure this is the case, consult your local dealer.

If you have manual recording levels, then try and keep the needle on the VU meter just below the red end of the scale when you record. If your deck uses a meter labelled from 0 to 10, then try to get the needle to sit about 3/4 of the way across the scale when you record.

Next, try moving the tape deck away from the TV or monitor, as the deck can try to record ambient electric fields created by the tube in these devices.

If these points don't improve matters, then try saving material on another tape deck and then loading it. If you are still encountering the same problem, then it is most probable that the computer is at fault, otherwise the tape deck is almost certainly at fault. If you have problems loading any software, then firstly make sure that any kind of tone control your deck may have is HIGH, or with as much treble as possible, and that the tape deck is, again, as far from any T.V. or monitor as possible.

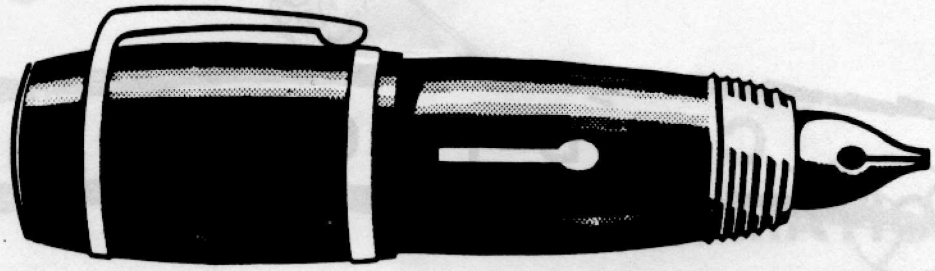
If you find you cannot load any pre-recorded software at all, but can save and load your own material, then be prepared to attack your tape deck with a screwdriver. On most tape decks there is a little screw, commonly known as the azimuth screw. If you look around the place that you load the tape into, you should (hopefully) find a small hole. Don't poke anything in it yet. This is the hole that allows access to the azimuth screw.

The azimuth screw basically controls the angle that the tape head sits at. This angle is known as the azimuth. When the software was recorded, the head sat at a (usually) specific azimuth. To load the software, the azimuth on your deck must match the azimuth it was recorded at. We won't go into explanations why this is, as it is uninteresting and is not really of global importance.

Now you know the gory details, you can attack your tape deck. Firstly get a small screwdriver (about 1mm across the blade will do). Now put in a pre - recorded tape, make sure you can hear what the deck plays, and press PLAY. Make sure you can hear part of the program and put the screwdriver in the hole above the azimuth screw,and make sure the screwdriver meets the screw. By experimenting, you should be able to make the tone of the buzz really clear. Just keep experimenting and testing to see if you can load the software.

IMPORTANT! Make sure you take the screwdriver out before you press STOP, otherwise you may cause a great deal of damage.

Hopefully this article should alleviate any tape problems you may have. If you cannot find a solution to your problem in here, then the only thing I can suggest is that you contact a computer specialist where you may find the nirvana of the tape based computing world.



CORRESPONDENCE

Lucky for some.

Dear Sir.

Thank you for informing me of the new start of the IEUG. It couldn't have come at a better time as mine is one of those 'Enterprises that is beginning to gather dust in a cupboard.

I believe I may be one of those who has postal Orders in your possession but, as I'm not sure of the numbers. I appreciate it will be almost impossible for you to return them. If you can return them I may be able to persuade the P.O. to refund the money.

Anyway, I'm happy to enclose a 15.00 cheque for a years membership and I look forward to finding out more about how the Enterprise ticks. I know a little Z80 programming (mainly Spectrum) and I hope that with a little help with understanding EXOS, I can contribute useful routines for the Enterprise. I though I was doing well with my DIY Dissasembler (written in Basic) until I got hold of a copy of the Enterprise Technical Manual.

All the best with the new IEUG.

Nick Green.

Nick was lucky I did manage to find his P.O. hopefully he was able to cash them in at the Post Office. If anybody else has any money outstanding with the old IEUG please write to me, there is no need for anybody to be short changed If I can help it. We don't have currently any articles on M/C programming lined up although we do have a Dutch user group magazine with articles covering such subjects. In the near future we aim to be translating them to English and publishing them, permission granted of cause.

Some one needs some answers.

Sir.

In reply to your invitation to subscribe to the IEUG. I am thinking of subscribing. HOWEVER in view of my previous experience, before subscribing, I require the answer to various queries.

1 Again your letter is full of ideas but lacking detail of when these ideas will be available ie, You promise a magazine. When will the 1st Issue be published and how often after the first issue.

2 Who is the IEUG ie who are the user group officers by name and position.

3 Does the IEUG have a bank account if so where?

4 Is the IEUG registered with a recognised computer club association if so who and registration number.

5 How can members contact the group for answering queries. Previously the telephone was always engaged (left of the hook??) and letters ignored.

6 Is the IEUG linked in any way to Enterprise Computers in West Germany.

Lastly, I enjoy compiling programs which I find difficult to do with the Enterprise due to lack of programing information (in particular using the "VIDEO:" facilities information is NON-EXISTANT). Is there information available, if so how is it obtained.

If these queries are answered to my satisfaction I will willingly send a subscription.

P Hughes

I don't wish to go over all the questions raised by Mr Hughes as I have already answered them personally, I will however use some of the points raised in his letter to clear up a some misconceptions and queries that judging by some of the other letters I have received are very common.Firstly who are we, by we I mean the people running the user group and producing the magazine. There are two of us, myself Tim Box and Eamonn O'leary. Eamonn is the art director in charge of the magazine, which is produced on a PC running VENTURA PUBLISHER a desktop publishing program. The trouble is I have the PC and he lives at the other end of a 1 hour Tube journey. His is also a general dogs body and provider of moral support (most invaluable). The rest which covers writing, arranging meetings, running and updating the membership database, letter and telephone answerer etc, etc is done by me (at the current time any way). The Group is not at the time of writing a member of any Computer Club Association. I aim by the time this publication reaches you to have forwarded an application form to the ACC. Any one wishing to contact the club can phone me at home on 01-346-8032 after 6pm and before 9pm (but not all at once). You can also write to me at the club address with queries. All letters will be attempted to be answered but as I hope you appreciate will not always be done immediately. A point to why the old IEUG phone always seamed to be engaged was because of the shear number of calls he use to answer. Lastly the IEUG is no way affiliated to Enterprise Germany.

You should be talking to each other.

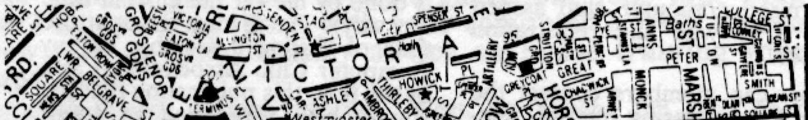
You should by now realise we want every body to be talking to each other. There are various ways of doing this, one being holding user group meetings. This has the disadvantage of only happening every two or three months. What is really needed is to get you talking to each other person to person. How do we intend to do this? By setting up a contact register. A contact register is (by my definition anyway) a list of users who have no objection to having there names and address being put on a list which can be integrated by other members. The list would be divided into regions of the counties and even countries if need be. All that is needed is you. If you want to be on the register all you have to do is write to us here at the I.E.U.G and say "I want to be on membership register" or some thing to that effect and your name and particulars will go down in our "membership register data base". Why not include subjects you are interested in like music or financial. Upon writing to us you will receive an up to date membership list, a membership list can also be supplied there after to keep you up to date with all the new members. There is only one other requirement and that is to do with the availability of the membership list. To obtain a copy you have to be a member, this way we can prevent the misuse of the service by those not 100 % serious. I look forward to a big response.

CLUB PAGE

LONDON SHOW; YOU'D BE A FOOL TO MISS IT !

Flushed with the success of the Last Alternative micro show in Birmingham, the organisers 'EMSOFT Ltd' are trying there luck in London. The last show received a lot of acclaim from the computing press as those of you who read ACE and Computer Shopper will know. The show this time has moved down south from the some what lesser known Aston Villa Sports and Social Center to the old haunting ground of the ZX Micro Fares, the New Horticultural hall in Victoria, London.

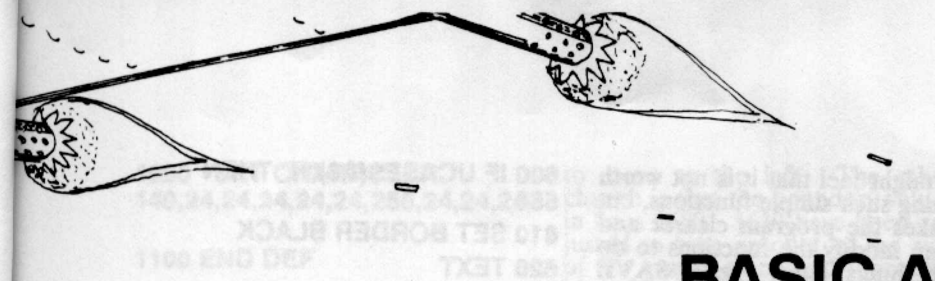
Along with ourselves there will also be a large number of stalls and stands representing the other heroes of a bygone computing age, in the guise of Dragon's, Oric's, Lynx's, QL's and Einstine's as well as a large number of retailers selling printers, disks, disk box's not to mention drives and monitors etc all at very reasonable prices. The show is readily accessible by tube and bus as shown in our map below. Entry is £2.00 at the door and the door in question opens at 10am. The date for this venue is the 1st of April....... NO honestly its no joke!



SATURDAY 1ST APRIL 1989
10AM-5PM
HORTICULTURAL HALL
GRAYCOAT STREET
LONDON SW1

LIBRARY IN NEED OF CURATOR.

With current shortage of software, we thought it would be a good idea to start a software library. The library, it is intended, will encourage the spread of the good software that we know a lot of you are capable of producing. To achieve it we are asking for your help. The first basic essential is obviously, software to be collected for the library, this should be runable from tape as the spread of disk drives has unfortunately halted due to lack of supplies. The software should also be accompanied with a file including the documentation, in WP format. The above requirements are of course not finalised and as you will have gathered not complete. The reason for this is the final requirement. We need some one to run it. The lucky candidate will be supplied with tape duplicating equipment or will be supplied with the tapes already processed. The main task is to compile and check the programs sent in by members as well as the obvious work of sending out tapes. If you think you fit the bill write to me Tim Box at the IEUG address enclosing your telephone number.



BASIC ANIMATION

By Peter Hiner

How to produce animated displays in Basic. That sounds rather ambitious! Indeed it is ambitious, especially if your mind immediately conjures up visions of the dazzling displays found in games like Sorcery or Cauldron. Well, do not expect to reach those dizzy heights, but we can at least experiment with some of the simpler techniques for moving things around the screen, providing joystick control, detecting collisions and so on. Even though the Enterprise Basic is notoriously slow, it is still possible to get things moving. If you compile your programs with Zzzip you will be surprised how fast you can go - try my version of Galaxians for size! The important thing is to get started.

In this article I will deal with animated Text screens, leaving the more complex problems of Graphics screens for another time. Don't be put off by the description "Text screens". You will find that we can do quite a lot with them. The first thing to note is that we shall not normally use the PRINT statement, but instead we shall POKE characters directly into screen memory. That is the technique used in machine code games. In Basic it is a bit slower to use POKE rather than PRINT, but when the program is compiled by Zzzip the speed improvement is fantastic.

Before we can POKE characters to screen we must find the area of RAM being used for video display. To do that we need a machine code patch

```
100 LET UNIQUE$ =
"SCREEN$"
110 LET DESIRED_
SPACE=100
120 LET SYZE = DESIRED_
SPACE+LEN(UNIQUE$)
130 IF VERNUM=2 THEN
140  LET T= PEEK(544) +
256*PEEK(545)-SYZE
150  FOR X=1 TO
LEN(UNIQUE$)
160   IF PEEK(X+T-1) ORD
(UNIQUE$(X:X)) THEN
170   ALLOCATE SYZE
180   POKE 542,PEEK(544)
190   POKE 543,PEEK(545)
200   CODE ="SCREEN$"
210   RUN
220  END IF 230 NEXT
240  LET X=X+T
250  POKE 540,X BAND 255
260  POKE 541,X/256
270 ELSE
280  ALLOCATE DESIRED_
SPACE
290 END IF
300 CODE FINDSEG = HEX$
("7D,6,3,F7,B,7A,F6,3F,7,7,6F,26
,0,C9")
310 CODE FINDVID = HEX$
("7D,6,3,F7,B,EB,CB,BC,CB,B4,
C9")
```

Most of these first lines are taken up by the familiar routine to get round the ALLOCATE bug. Note that the variable name SIZE has been corrected to SYZE, to distinguish it from the built-in function and so to avoid confusing the Zzzip compiler. Line 300 finds the memory segment in which the text screen is stored, and line 310 finds the address of the first character on screen (top left corner). Just to check that everything is working correctly, we will try a simple routine to fill the screen with letters.

```
320 TEXT 40
330 LET SEG = USR
(FINDSEG,102)
340 LET VID=USR
(FINDVID,102)
350 FOR X=65 TO 90
360 FOR Y=VID TO VID+960
370 SPOKE SEG,Y,X
380 NEXT Y
390 NEXT X
```

I do not propose to explain the operation of the machine code patch in detail, but I do want to make sure that the techniques used in lines 320 to 390 are clearly understood. Line 320 clears the Text page, making sure that lines are displayed in the expected order. Lines 330 and 340 call the machine code patch. Now we can POKE any character into any position on screen using the SPOKE command as in line 370, where SEG selects the required segment of memory, Y is the position on screen and X is the character. Line 350 varies the value of X from letter A to letter Z. Line 360 varies the value of Y from VID (top left corner) to VID+960 (bottom right corner). You will note that there are 24 horizontal lines on the standard Text page, with 40 characters per line (24*40=960). So to address a vertical column of screen positions, you would just keep on adding 40 to the value of Y. You can try changing line 360 to:

```
360 FOR Y=VID TO VID+960
STEP 45
```

Now we will delete those lines 320 to 390 and carry on with our main program. The next bit is straightforward housekeeping. I suggest that you keep to the line numbers I have used and include the REM lines, which we will fill in later. Some of the variables declared in line 320 are for future use.

```
320 NUMERIC I,J,N,B1,B2,B3,
B4,B5,B6
330 SET KEY CLICK OFF
340 !
350 TEXT 40
360 SET £102:PALETTE
CYAN,BLACK,CYAN,RED,0,0,0,
370 SET BORDER CYAN
380 LET SEG=USR
(FINDSEG,102)
390 LET VID=USR
(FINDVID,102)
400 LET X=10:LET Y=5:LET
NEWPOS=210
410 !
420 SET STATUS OFF
430 !
440 !
450 !
```

Now we come to the most important section. It is the main loop which determines the sequence of events in our display or game.

```
460 DO
470  SPOKE SEG,VID+
NEWPOS,42
480  CALL MOVJOY
490 LOOP
```

Line 470 puts a star on screen at a position determined by the value of NEWPOS. Line 480 calls MOVJOY to detect movement of the joystick and calculate a new value for NEWPOS. MOVJOY includes some checks to avoid going outside the screen area or jumping from one side to the other.

```
1500 DEF MOVJOY
1510   LET J=JOY(0)
1520   IF J BAND 1 THEN LET
X=X+1
1530   IF J BAND 2 THEN LET
X=X-1
1540   IF J BAND 4 THEN LET
Y=Y+1
1550   IF J BAND 8 THEN LET
Y=Y-1
1560   !
1570   !
1580   IF X THEN LET X=0
1590   IF X39 THEN LET X=39
1600   IF Y THEN LET Y=0
1610   IF Y23 THEN LET Y=23
1620   LET NEWPOS=Y*40+X
1630   !
1640 END DEF
```

SAVE the program so far and then RUN it. You should find that moving the joystick leaves a trail of stars. However for serious animation we want to erase our trail, preferably restoring the character previously displayed. To do that we must keep a record of both the previous position and the character displayed there. We will turn lines 460 onwards into a general purpose loop, calling up a sequence of functions which can be defined to suit different purposes.

```
460 DO
470   LET OLDPOS=NEWPOS
480   CALL SAVOLD
490   CALL DRAWNEW
500   CALL MOVJOY
510   CALL SCORE
520   CALL DRAWOLD
530 LOOP
```

Now we need to define some new functions.

```
800 DEF SCORE
810   ! NOTHING HERE YET
820 END DEF
1800 DEF SAVOLD
1810   LET B1=SPEEK
(SEG,VID+OLDPOS)
1820 END DEF
2000 DEF DRAWNEW
2010   SPOKE
SEG,VID+NEWPOS,42
2020 END DEF
2800 DEF DRAWOLD
2810   SPOKE
SEG,VID+OLDPOS,B1
2820 END DEF
```

You might feel that it is not worth defining such simple functions, but it makes the program clearer and we can modify the functions to do other things later. Again SAVE the program so far and then RUN it. You should now have a single star which you can steer round the screen. It is worth making sure you understand exactly what is going on in this version of the program, as it contains the essential ingredients for animation.

You will notice that the star flickers. That is because for a large proportion of the time it is blank. You could cure this by changing line 520 so that the star is only deleted if there is a change in its position:

```
520 IF NEWPOSOLDPOS
THEN CALL DRAWOLD
```

However that is not a general solution and would cause all sorts of problem in a more complex program. The real solution is to change the ratio of time displayed to time blank. Since we have nothing else we want to do in this program we could simply waste some time by adding a delay, for example in the SCORE function:

```
810 FOR I=1 TO 100
815 NEXT
```

Obviously this would have a disastrous effect on the speed of a Basic program, but it would be quite acceptable in a compiled program. In fact you could change the length of the delay to provide a variable speed control.

Now we will make some more changes and additions to turn our program into a game (originally published in Personal Computer World).

Change lines 410,440,450 and 530, and add lines 540 to 650.

```
410 LET AM=50:LET
SC=0:LET FF=25
440 TIME "00:00:00"
450 PRINT £102,AT 1,1:"ammo
left      hits      fuel left  ";
530 LOOP UNTIL(FF=0)
OR(AM=0) OR(SC=50)
540 IF FF=0 THEN PRINT
£102,AT 3,1:"Sorry - you ran
out of fuel";
550 IF AM=0 THEN PRINT
£102,AT 3,1:"Sorry - you ran
out of ammo";
560 IF SC=50 THEN PRINT
£102,AT 3,1:"Well done - shot
him down in flames";
570 PRINT £102,AT 5,1:"Want
another go ? (y/n) ";
580 GET A$
590 IF UCASE$(A$)="Y" THEN
350
600 IF UCASE$(A$)"N" THEN
580
610 SET BORDER BLACK
620 TEXT
630 CLEAR FONT
640 SET KEY CLICK ON
650 END
```

The program will not do anything useful yet, but note in passing that you can PRINT to screen via channel 102, provided that you avoid carriage returns (hence the semicolon at the end of lines 540 to 570). Now we will define the function SCORE, so that it decreases the amount of fuel every 5 seconds, and reports the state of play every cycle.

```
800 DEF SCORE
810   IF TIME$(8:8)="5" THEN
820     LET FF=FF-1
830     TIME "00:00:00"
840   END IF
850   PRINT £102,AT 1,10:AM;
860   PRINT £102,AT 1,21:SC;
870   PRINT £102,AT 1,36:FF;
880 END DEF
```

Next we will modify line 340 to:

```
340 CALL SETFONT
```

In SETFONT we will define some characters for drawing an aeroplane and a gun- sight. Note that, for example, in line 1010 we SET CHARACTER 131, but when we use that character in a SPOKE statement we shall refer to it as number 3. What we are effectively doing is subtracting 128 from the character number so as to make it appear in the normal ink colour. Later on you will see that in lines 3130 to 3160 we POKE character 129, which causes gunfire to appear in red.

```
1000 DEF SETFONT
1010   SET CHARACTER
131,1,1,1,1,1,1,1,1,1
1020   SET CHARACTER
132,128,128,128,128,128,128,12
8,128,128
1030   SET CHARACTER
134,0,0,0,0,255,0,0,0,0
1040   SET CHARACTER
135,24,24,24,24,24,24,24,24,24
1050   SET CHARACTER
136,60,102,195,129,129,129,195
,102,60
1060   SET CHARACTER
137,0,0,0, ,0,0,0,0,255
1070   SET CHARACTER
138,255,0,0,0,0,0,0,0,0
1080   SET CHARACTER
139,24,60,126,255,255,255,255,1
26,60
```

```
1090  SET CHARACTER
140,24,24,24,24,24,255,24,24,24

1100 END DEF
```

Our previous function SAVOLD only stored one character value. Now we want to store all the characters covered by an aeroplane, so change SAVOLD to:

```
1800 DEF SAVOLD
1810   LET N=VID+OLDPOS
1820   LET B1=SPEEK
(SEG,N-2)
1830   LET B2=SPEEK
(SEG,N-1)
1840   LET B3=SPEEK
(SEG,N)
1850   LET B4=SPEEK
(SEG,N+1)
1860   LET B5=SPEEK
(SEG,N+2)
1870   LET B6=SPEEK
(SEG,N-40)
1880 END DEF
```

Similarly we must change DRAWNEW to display an aeroplane instead of a star, and DRAWOLD to replace more characters:

```
2000 DEF DRAWNEW
2010   LET N=VID+NEWPOS
2020   SPOKE SEG,N-40,12
2030   SPOKE SEG,N-2,6
2040   SPOKE SEG,N-1,6
2050   SPOKE SEG,N,11
2060   SPOKE SEG,N+1,6
2070   SPOKE SEG,N+2,6
2080 END DEF
2800 DEF DRAWOLD
2810   LET N=VID+OLDPOS
2820   SPOKE SEG,N-2,B1
2830   SPOKE SEG,N-1,B2
2840   SPOKE SEG,N,B3
2850   SPOKE SEG,N+1,B4
2860   SPOKE SEG,N+2,B5
2870   SPOKE SEG,N-40,B6
2880 END DEF
```

Now we will fill in the blank lines 1560, 1570 and 1630 (in the function MOVJOY), to generate random evasive movements of the aeroplane and to detect operation of the fire button. We need to make some other changes to MOVJOY and therefore the whole function is listed again here for clarity. You will note that the directions of movement have been reversed in lines 1520 - 1550. The reason for this is that the screen is intended to show an enemy plane trying to escape from your guns. If you move your plane to the right, the enemy plane will appear to move to the left. The other change is that the boundary values in lines 1580 - 1610 have been adjusted to take account of the size of the enemy plane and toprevent bits of it wrapping round the edge of the screen.

```
1500 DEF MOVJOY
1510   LET J=JOY(0)
1520   IF J BAND 1 THEN LET
X=X-1
1530   IF J BAND 2 THEN LET
X=X+1
1540   IF J BAND 4 THEN LET
Y=Y-1
1550   IF J BAND 8 THEN LET
Y=Y+1
1560   LET
N=INT(RND(3))-1:LET
X=X+N
1570   LET N=INT(RND(3))
-1:LET Y=Y+N
1580   IF X THEN LET X=2
1590   IF X37 THEN LET X=37
1600   IF Y THEN LET Y=1
1610   IF Y23 THEN LET Y=23
1620   LET NEWPOS=Y*40+X
1630   IF (J BAND 16) THEN
CALL FIRE
1640 END DEF
```

Then we define the FIRE function:

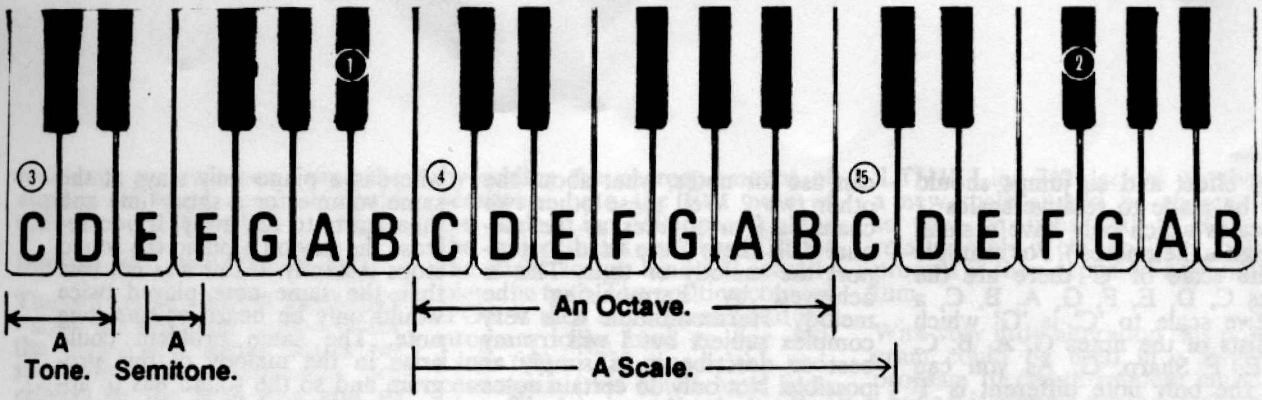
```
3000 DEF FIRE
3010   LET J=SPEEK
(SEG,VID+420)
3020   IF J=11 THEN
3030     LET N=5
3040     LET SC=SC+5
3050   ELSE IF J=6 OR J=12
THEN
3060     LET N=2
3070     LET SC=SC+2
3080   ELSE
3090     LET N=2
3100   END IF
3110   LET AM=AM-1
3120   FOR I=1 TO N
3130     SPOKE SEG,VID+
379,129
3140     SPOKE SEG, VID+
381,129
3150     SPOKE SEG, VID+
459,129
3160     SPOKE SEG, VID+
461,129
3170     SPOKE SEG, VID+
379,32
3180     SPOKE SEG, VID+
381,32
3190     SPOKE SEG, VID+
459,32
3200     SPOKE SEG, VID+
461,32
3210   NEXT
3220 END DEF
```

Finally we add a gun-sight as follows:

```
430 CALL SIGHT
2500 DEF SIGHT
2510   FOR I=257 TO 263
2520     SPOKE SEG,VID+I,9
2530   NEXT
2540   FOR I=577 TO 583
2550     SPOKE SEG,VID+I,10
2560   NEXT
2570   FOR I=335 TO 495
STEP 40
2580     SPOKE SEG,VID+I,3
2590   NEXT
2600   FOR I=345 TO 505
STEP 40
2610     SPOKE SEG,VID+I,4
2620   NEXT
2630   FOR I=300 TO 540
STEP 40
2640     SPOKE SEG,VID+I,7
2650   NEXT
2660   FOR I=416 TO 424
2670     SPOKE SEG,VID+I,6
2680   NEXT 2690   SPOKE
2690   SPOKE  SEG,VID+420,8
2700 END DEF
```

Be sure to SAVE the program before trying to RUN it. If all is well, you should find yourself piloting a Spitfire on the tail of an Me109. You must register 50 hits to shoot him down, but don't run out of fuel or ammo. Hits on his fuselage score higher than hits on the wings. I will leave you to add the instructions and other details to the program. One last point. The program runs at quite a respectable speed in Basic. You can try compiling it, but then it will run so fast that it is almost impossible

(unless you add a delay into the SCORE function). However this reserve of speed will be vey useful when we try something more complex.



1 - B Flat.
2 - F Sharp.
3 - Pitch 25 on the Enterprise.
4 - Pitch 37 on the Enterprise.
5 - Pitch 49 on the Enterprise.

AUTO MUSIC

By Anthony Shaw

The 'Auto music' program originated while I was developing a quiz program. I wanted constant, background music, but I did not want to type in 30K of data (sort of like digging a well with a teaspoon). So I decided to make my trusty Enterprise do the work for me. I'm sure many of you must have tried this by playing random notes, and many of you must have come up with the same disgusting sounds. Music, or so I am told, is very mathematical (is this why I am doing maths at 'A' level ?) and rules have to be followed to produce any sound that is to be pleasurable (?!!) to the ear. I have tried to apply the main rules in my program and so produce a fairly acceptable noise.

There are many rules in music, too numerous to mention, but the basic ones are included. If you are musically inclined then some of the other rules could be applied or the rules in the program could be made stricter, but in an attempt not to confuse too many people I will leave that up to your ingenuity.

I presume, rightly or wrongly, that majority of you will, at some time in your life, have had a tinkle on a piano or played some instrument. If you have, then you will realise that playing notes (or sounds) in any order does not make a pleasurable tune, and only certain notes played after each other sound alright. To fully understand the basic concepts in this program I will have to give a short (!) lesson in music - for those who have a basic knowledge of music skip to page 16 of this article (only joking).

All music is created from twelve notes - 'C' to 'B' - pitches 37 to 48 on the Enterprise. These twelve notes, or pitches, are separated by what is known as a 'semitone', a semitone is the smallest move possible on a piano. For example, note 'E' to note 'F', see diag. 1. Two 'semitones' make one 'tone' , note 'C' to note 'D', see diag. 1. These twelve notes or semitones make up an 'octave'. The octave then repeats itself just at a higher pitch. For example, on the Enterprise, pitches 37 and 49 are both the note 'C' but pitch 49 is higher than 37 but still sounds the same. Pitch 49 is said to be one octave higher than pitch 37. (Do you ever get the feeling you're confusing people more the simpler you get ?????) Therefore applying this rule pitch 25 is one octave below pitch 37, all you are doing is adding or subtracting twelve each time, because each pitch number om the Enterprise is one semitone.

To summarise - an octave consists of twelve semitones which are the shortest distance that can be moved on a piano. As I have said before only certain sequences of these notes will produce an acceptable sound. An acceptable sequence of these notes is called a 'Scale'. A 'Scale' consists of eight notes (the first being repeated). For example the scale of 'C' consists of the notes C, D, E, F, G, A, B, C. This is the simplest scale because on a piano only the white notes are used. If these notes are played one after the other the sound is pleasant to the ear.

From the scale of 'C' we can work out the semitone jumps between the notes, ie. from 'C' to 'D' there is a jump of two semitones, from 'E' to 'F' there is a jump of one semitone. The sequence of semitone jumps in this scale is 2 , 2, 1, 2, 2, 2, 1. If this sequence of semitones is applied to any note then a pleasurable sound will occur. For example applying this sequence to the note 'G' gives the following notes - G, A, B, C, D, E, F sharp (the black note after 'F' [see diag. 1]), G - this is the scale of 'G'.

Therefore jumping to any note in this scale will be pleasant to the ear although jumps to the second and third notes are found to be more pleasant sounding (For the music buffs out there I know a neapolitan sixth to a second inversion tonic is very nice, but these can be added by your goodselves). So we now have our first set of rules (First set ? You mean there's more ?)

So by choosing a starting note we have eight notes with which to play (excuse the pun !), "but won't this get boring ? " I hear you cry. Well, in a word , yes ! Starting your scale on a different note, however, gives up to twelve scales (simply, there are actually fifteen). I shall try to explain the rest in terms of scales starting on different notes, and I hope a won't have to go into the key system (phew !). Just as jumping from a note to any other can cause an unpleasant sound, jumping from one scale to another can have the same effect and so jumps should only be made to relative scales (or scales which only have a small change in their notes). For example in the scale of 'C' there are the notes C, D, E, F, G, A, B, C, a relative scale to 'C' is 'G' which consists of the notes G, A, B, C, D, E, F Sharp, G. As you can see, the only note different is 'F sharp' and so the transition is acceptable. Another related scale is the scale of 'F' this scale consisits of the notes F, G, A, B Flat (The black note before 'B' on the piano [see diag. 1]. As you have probably worked out a 'Sharp' raises the note by a semitone and a 'Flat' lowers the note by a semitone), C, D , E, F. This scale is also only one note different from the scale of 'C'. These two scales - 'G' and 'F' - start on the fifth and fourth notes (respectively) of the scale of 'C'. It is found that the scales starting on the fourth and fifth notes of any particular scale will only be one note different and so these are ideal scales to move between.

How about another summary ? Well you're going to get one anyway. There are twelve semitones in an octave. If, starting on any note, the sequence of semitones (2, 2, 1, 2, 2, 2, 1, see line 126 0 in the program) is applied then a pleasent sequence of notes will be formed called a scale (although this is not always true because Bob Dylan has just come on the radio and somehow he manages to defeat the pleasant part of it - all letters from the Bob Dylan fan club to Neil Blaber, P.O. BOX 13,etc), by taking the fourth or fifth note of this scale and using this as the first note of another scale a pleasurable transition can be made between scales and so access to all the notes of the octave.

Only 'Minor' scales and harmony to do now (is that all ?!). A good time to have a coffee and digest the first half before the struggle ahead.

I have already discussed using the sequence of semitones 2, 2, 1, 2, 2, 2 , 1. This creates a happy, lively sound and a scale using this sequence is called a 'major' scale (at two in the morning I can't remember where 'major' comes from so no complaints please) There are also 'minor' scales, these use a different semitone pattern and have a sadder, more mournful sound. The semitone sequence for the 'minor' scales is - 2, 1, 2, 2, 1, 3, 1 (see line 1280 in the program). So we have the relative major keys and now relative minor keys. The relative minor is three semitones down from the first note of the major scale. For example the relative minor to the 'C' major scale is 'A' minor. This now doubles the amount of scales we can use and so makes the music more interesting.

This only accounts for one of the three channels that the Enterprise can use for notes, what about the other two ? Well these other two channels (or sources as the manual calls them) are used to support the melody or tune. This is achieved by 'harmonising' the melody. Harmonisation is a very complex subject but I will try my best to describe it as simply as possible. Not only do certain notes following each other sound either good or bad but certain notes played at the same time can sound good or bad. Try playing 'C', 'E' and 'G' on your Enterprise at the same time (pitches 37, 41 and 44). These notes are pleasant to listen to, whereas try 'E', 'F' and 'F Sharp' (pitches 41, 42, 43). Horrible isn't it ?!!!!. The first, third and fifth notes sound pleasant and are called the 'Tonic Triad' of a scale. I use this 'Tonic Triad' as harmony, or supporting, notes.

So now we have the pitches sorted out (at least I hope so !!!), but a piece of music is made from more than pitches. An integral part of the music is the rhythm (being a percussionist I know this). Without going into bars and time signatures, basically, if there is a constant pulse running through a piece or the duration or length of the notes always add up to the same value, then the music will sound alright. If you look at lines 1160 to 1240 in the program and add up the values in each line (the number 255 is not part of this, it is just used as an end of data marker) you will find that each data statement adds up to twenty eight. Therefore there is some consistancy in the rhythms of the program.

Basically the program selects a scale, plays two notes from the triad and then over the top of this plays notes from the scale with jumps of not more than four semitones. The rhythms are selected from preset groups all adding up to the same amount. After two of these groups has been played a new scale , from a list of related scales, is chosen and the process repeated. (simple huh !).

I will now explain how I achieved this in my program. The program is split into six parts as follows.....

SECTION	LINE NOs	DESCRIPTION
1	100 - 340	Initialisation
2	350 - 460	Harmony notes
3	470 - 600	First part of melody.
4	610 - 720	Second part of melody
5	730 - 830	Choose new rhythm patterns and new scales
6	840 - 1280	Data.

Firstly, in lines 110 and 120, the two envelopes are defined. Envelope one (line 110) is used for the melody notes, or the tune. An envelope defines the change of the pitch and volume as the note is played. For example, a flute stays at the same volume as it is played, whereas a piano only stays at the same volume for a short time and then starts to die away. If you release the key on a piano the sound stops instantly. If this was not true then the same note played twice would only be heard as one long note. The same problem could arise in the melody of this program and so the sound has to appear quickly and disappear quickly. Therefore envelope number one goes to full volume instantly, stays there for a fraction of a second then dies quickly to a quiet sound. The sound then slowly disappears altogether. This means that if the same note is played twice, although the pitch won't change, the volume will rapidly rise again and so a distinction between the two will be heard. The opposite is true however of envelope two, this is used for the harmony notes which play continually below the melody. If these harmony notes were to die away quickly there would be no harmony for the rest of the melody. These notes, then, have to stay at a constant volume to provide harmony for all of the tune.

After making sure the computer is not trying to play other sounds by using the CLEAR SOUND command in line 130, I start to read in the data. In line 140 the arrays are set up, NOTES holds the relevant information about the scales, it holds six pieces of information about each of the thirty scales (fifteen major, fifteen minor). The first three pieces of information hold the triad for that scale (this being used in the harmony), the next three hold the relative scales (the relative minor, the fifth note of the scale and the fourth note of the scale). The triad is stored as pitches used by the Enterprise (in 'C' the triad is 37 - 'C' 41 - 'E', 44 - 'G', see line 850). The relative scales, however, are stored as their position in 'NOTES', eg. In 'C' the fifth note is 'G' which is stored in the second place of 'NOTES' - ie. NOTES(2), therfore the fifth piece of data in line 850 is the number two. If this number was three then it would relate to the scale of 'D'.

The next array (see page 72 in the manual) is called LEG; LEG stores the groups of note values, or rhythms. There are eight of these groups which can be found in lines 1170 to 1240. You will notice that at the end of each of these lines there is the number 255; this is used as an end of data marker (see line 250). Each group of rhythms have different amounts of numbers and so to read them in an easy way without having a different routine to read each separate one I used the number 255 so the computer could tell when the data had

finished (see lines 200 to 260).

The two arrays - MAJS and MINS - store the semitone sequences for the major and minor scales respectively. These sequences are relative to the first note and so the major sequence is 0, 2, 4, 5, 7, 9, 11, 12 and not 2, 2, 1, 2, 2, 2, 1, but if you examine the former list the jumps between the numbers are found in the latter list. The same applies to the minor sequence. As I have previously said, the program selects a scale, plays two notes from the triad, and then over the top of this plays notes of the scale with jumps of not more than four semitones. The rhythms are selected from preset groups all adding up to the same amount. After two of these groups have been played, a new scale, from a list of related scales, is chosen and the process repeated. This can now be achieved because in store we have the scales, the triads, the semitone jumps, the preset rhythms and the related scales. So it is now just a case of looking these items up in the arrays where they are held.

After reading in all the data into the apropriate arrays, a random scale is chosen and held in the variable 'KEY', then two variables that store the rhythms to be used - LTH and LTH1 - are initialised. In line 340 the variable 'OCTAVE' is used to store the octave number. If the program moves off the top of the scale then the octave is increased or off the bottom then the octave is decreased.

The main loop starts at line 380 by choosing the notes of the triad to be played by each channel (or source) of the Enterprise. These notes are held in A, A1 and A2. Lines 440 to 460 actually play the triad, lines 440 and 450 play the harmony notes and line 460 plays the first melody note (which is always a note from the triad for easy movement between scales).

The variable 'POS' in line 470 is used as a counter, the computer then knows which rhythm to use next. In line 480 'PITP' is assigned with one; 'PITP' is the note in the scale to be played, it is assigned with one so the melody always starts on the first note of the scale. Lines 490 to 600 repeat until the end of the rhythm is found (using the end of data pointer 255). Line 500 chooses the number of semitones to move (-1 to 3), this is then stored in the variable 'PITS'. This value is then added onto 'PITP' in the next line. Lines 550 to 550 check to see if the note has left the limits of the scale. If it has, it is corrected and the octave is moved up or down. If the octave is above two or it is below minus two then it becomes zero or two respectively. Lines 560 and 570 change the position in the scale ('PITP') to an actual note by adding the correct amount of semitones onto the first note of the scale. Line 560 does this for the major scales and line 570 for the minor scales. Line 580 actually plays the note. The position counter 'POS' is then incremented and the program loops back to line 490. Lines 610 to 720 repeats this process again (this could have been placed in a subroutine and called twice but this is a problem that comes with writing spaghetti programs and not planning them out).

Two new rhythms are chosen in line 730 and stored in 'LTH' and 'LTH1'. Line 740 decides whether a new scale should be chosen if so then in lines 760 to 820 a relative scale is chosen. The whole process is then repeated ad infinitum.

With very little alteration this program could be used in a larger program to create random music. I hope it is of some use to somebody out there !

GLOSSARY

MUSICAL SOUNDS are named, in ascending order, from the first seven letters A - G, and these are repeated to represent the same notes at a higher or lower level.

OCTAVE (Eight) is the term given to the next sound, either above or below, which has the same letter name.

SEMITONE is the smallest distance between two notes on the keyboard.

TONE is two semitones.

SHARP raises a note one semitone in pitch.

FLAT lowers a note one semitone in pitch.

SCALE (Latin 'SCALA' a ladder) is an alphabetical succession of sounds ascending or descending from a starting note.

TRIAD the first, third and fifth notes of a scale.

PULSE The steady throb one could clap to when listening to music.

```
100 PROGRAM "AUTO_MUSIC"

110 ENVELOPE NUMBER
1;0,63,63,1;RELEASE;0,0,0,5;0,-30,-30,10;0,-200,-200,100
120 ENVELOPE NUMBER 2;0,63,63,2;RELEASE;0,0,0,150
130 CLEAR SOUND
140 DIM NOTES(30,6),LEG(8,11),MAJS(8),MINS(8)
150 FOR F=1 TO 30
160   FOR G=1 TO 6
170     READ NOTES(F,G)
180   NEXT
190 NEXT
200 FOR F=1 TO 8
210   LET G=0
220   DO
230     LET G=G+1
240     READ LEG(F,G)
250   LOOP UNTIL LEG(F,G)=255
260 NEXT
270 FOR F=1 TO 8
280   READ MAJS(F)
290 NEXT
300 FOR F=1 TO 8
```

```
310  READ MINS(F)
320 NEXT
330 LET KEY = RND(30) + 1:LET LTH = 8:LET
LTH1 = 5
340 LET OCTAVE = 0
350 DO
360  RANDOMIZE
370  LET A = RND(3) + 1
380  LET A1 = RND(3) + 1
390  IF A1 = A THEN 380
400  IF A = 1 AND A1 = 2 OR A1 = 1 AND A1 = 2 THEN
LET A2 = 3
410  IF A = 1 AND A1 = 3 OR A1 = 1 AND A = 3 THEN
LET A2 = 2
420  IF A = 2 AND A1 = 3 OR A = 3 AND A1 = 2 THEN
LET A2 = 1
430  IF A2 = A1 OR A2 = A THEN 400
440  SOUND PITCH NOTES(KEY,A),DURATION
56,ENVELOPE 2,SYNC 2,STYLE 0,LEFT 75,R IGHT
25
450  SOUND PITCH NOTES(KEY,A2),DURATION
56,ENVELOPE 2,SOURCE 1,SYNC 2,LEFT 25
,RIGHT 75
460  SOUND PITCH NOTES(KEY,A1) +
(12*OCTAVE),DURATION LEG(LTH,1),ENVELOPE
1,SOU RCE 2,STYLE 0,LEFT 200,RIGHT 200,SYNC 2
470  LET POS = 2
480  LET PITP = 1
490  DO UNTIL LEG(LTH,POS) = 255
500    LET PITS = INT(RND(4)-1)
510    LET PITP = PITP + PITS
520    IF PITP8 THEN LET PITP = (PITP-8):LET
OCTAVE = OCTAVE + 1
530    IF PITP THEN LET PITP = (8 + PITP):LET
OCTAVE = OCTAVE-1
540    IF OCTAVE2 THEN LET OCTAVE = 2
550    IF OCTAVE2 THEN LET OCTAVE = 0
560    IF KEY THEN LET
PACT = NOTES(KEY,1) + MAJS(PITP)
570    IF KEY = 16 THEN LET
PACT = NOTES(KEY,1) + MINS(PITP)
580    SOUND PITCH PACT + (12*OCTAVE),
DURATION LEG(LTH,POS),ENVELOPE 1,SOURCE 2
,STYLE 0,LEFT 200,RIGHT 200
590    LET POS = POS + 1
600  LOOP
610  LET POS = 1
620  DO UNTIL LEG(LTH1,POS) = 255
630    LET PITS = INT(RND(4)-1)
640    LET PITP = PITP + PITS
650    IF PITP8 THEN LET PITP = 8-(PITP-8)
660    IF PITP THEN LET PITP = 1-PITP
670    IF KEY THEN LET
PACT = NOTES(KEY,1) + MAJS(PITP)
680    IF KEY = 16 THEN LET
PACT = NOTES(KEY,1) + MINS(PITP)
690    IF LEG(LTH1,POS + 1) = 255 THEN LET
PACT = NOTES(KEY,1)
700    SOUND PITCH PACT + (12*OCTAVE),
DURATION LEG(LTH1,POS),ENVELOPE 1,SOURCE
2,STYLE 0,LEFT 200,RIGHT 200
710    LET POS = POS + 1
720  LOOP
730  LET LTH = RND(8) + 1:LET
LTH1 = RND(8) + 1
740  IF RND.4 THEN
750    IF KEY AND RND.8 THEN LET
A = 4:GOTO 800
760    IF KEY THEN LET A = RND(2) + 5
:GOTO 800
770    IF RND.7 THEN LET A = 5:GOTO 800
780    IF RND.7 THEN LET A = 6:GOTO 800
790    LET A = 4
800    LET KEY = NOTES(KEY,A)
810    LET LTH = 8:LET LTH1 = RND(8) + 1
820  END IF
830 LOOP
840 ! MAJOR
850 DATA 37,41,44,16,2,9 ! C
860 DATA 44,48,51,17,3,1 ! G
870 DATA 39,43,46,18,4,2 ! D
880 DATA 46,50,53,19,5,3 ! A
890 DATA 41,45,48,20,6,4 ! E
900 DATA 48,52,55,21,7,5 ! B
910 DATA 43,47,50,22,8,6 ! F
920 DATA 38,42,45,23,12,7 ! C
930 DATA 42,46,49,24,1,10 ! F
940 DATA 35,39,42,25,9,11 ! Bb
950 DATA 40,44,47,26,10,12 ! Eb
960 DATA 45,49,52,27,11,13 ! Ab
970 DATA 38,42,45,28,12,14 ! Db
980 DATA 43,47,50,29,13,15 ! Gb
990 DATA 36,40,43,30,14,5 ! Cb
1000 ! MINOR
1010 DATA 46,49,53,1,17,24 ! A
1020 DATA 41,44,48,2,18,16 ! E
1030 DATA 36,39,43,3,19,17 ! B
1040 DATA 43,46,50,4,20,18 ! F
1050 DATA 38,41,45,5,21,19 ! C
1060 DATA 45,48,52,6,22,20 ! G
1070 DATA 40,43,47,7,23,21 ! D
1080 DATA 47,50,54,8,27,22 ! A
1090 DATA 39,42,46,9,16,25 ! D
1100 DATA 44,47,51,10,24,26 ! G
1110 DATA 37,40,44,11,25,27 ! C
1120 DATA 42,45,49,12,26,28 ! F
1130 DATA 35,38,42,13,27,29 ! Bb
1140 DATA 40,43,47,14,28,30 ! Eb
1150 DATA 45,48,52,15,29,8 ! Ab
1160 ! lengths
1170 DATA 7,7,7,7,255
1180 DATA 14,7,7,255
1190 DATA 7,14,7,255
1200 DATA 7,7,14,255
1210 DATA 14,14,255
1220 DATA 21,7,255
1230 DATA 7,21,255
1240 DATA 28,255
1250 ! major semitones
1260 DATA 0,2,4,5,7,9,11,12
1270 ! minor semitones
1280 DATA 0,2,3,5,7,8,11,12
```

PROG

AUTH
Andre

PUBL

Pain
for
but
been
has
in a
ing

T

reviews

PROGRAM. PAINTBOX.

AUTHORS. Andrew Burnham & Andrew Richards.

PUBLISHER. BOXSOFT.

Paintbox has been around for well over a year now but up till now has never been reviewed. So how has the program faired in all that time, is it showing its age?

To begin with what is Paintbox? Paintbox is a combination of a mouse, mouse interface and an art package. The art package is an all singing all dancing affair, boasting many features a lot of 16 bit machines art programs lack.

The package comes in a small polystyrene tray wrapped in a two tone blue printed cardboard wrapper, unfortunately it didn't stand up to the handling of the British postal service, even when wrapped in a large jiffy bag. The corners which presumably where nice and square now have a certain rounded look, this is a shame as it is one of the nicest looking packaging I've seen for any Enterprise product. The contents of the packaging though, did stand up to the rigorous PO treatment very well and all parts working first time.

Looking at each piece of the contents in turn:-

The mouse was designed for use with the Commodore 64/128 series of micros and as such has some advantages and disadvantages when used with the Enterprise. The main advantage is the way the mouse is read to establish its movements. All the work of monitoring movement is done by the on board 4 bit microprocessor, this saves a great deal of time for the Enterprises Z80 which would normally be interrupted every time you moved. This constant interrupting would slow any decent program down to a snails pace or restrict the speed that the mouse could be moved around, as any one using an AMX mouse for the 8 bit Amstrads will testify. The second major advantage is the way in which it can be interfaced to the Enterprises ports with minimum cost. Once again looking at an Amstrad or Spectrum will reveal a large and potentially costly interface necessary to enable there mice to work. There are as I said disadvantages but the only one worth talking about is the low resolution of only 100 points per inch. This may suit the Commadore with its low resolution screen but not the likes of the Enterprise that can display up to and over 720 points horizontally, it takes a lot of desk to move from one side of thby Boxsoft that a new mouse driver has just been written to over come this problem by incorporating within it an acceleration detector, that increases the pointer movement the quicker you move the mouse. I cannot comment on its improvements to the use of the mouse as I have not seen it yet but it does sound as if it would work. The mouse interface comes in two parts, the main piece is about three inches long and 1 inch wide. When in use it is plugged into "Control port one", it also houses the socket the mouse is connected to. In addition to this interface there is an extra lead necessary for the unit to work, this lead goes between the Serial port and the main mouse interface. At first, I thought, why have the lead detachable, that was till I realised that with the lead plugged in its a mouse interface and with out its a joystick interface, plugging the two together obviously switches it between the two. I had no trouble in assembling the whole lot to my Enterprise and apart from a little jiggling now and then it worked flawless through out.The software.

Paintbox the program comes on cassette, unless disk is requested, there is no difference between the two, except the price, the disk version being 2.00 more expensive. Since I was going to use it primarily from disk I resisted the temptation to load it straight in from tape, instead spent the next ten minutes copying it over to my 3 1/2 disk system. This process was a totally painless affair, just requiring me to make note of the file names, rewinding the tape and copying them over with the appropriate file names. Upon loading after being shown a short loading screen (at least when loading from disk) you are presented with display divided into two main parts, the 'COMMAND AREA' and the 'CANVAS'.

The Command area occupies the top half of the screen with all the 'COMMAND', 'STIPPLE', 'PALETTE' and 'BRUSH' icons held within it. Initially the 'POINTER'

The first thing you want to do is try it out, to do so you have to move into the canvas area, pulling the mouse down moves the Pointer to the intersection between the two screens where upon it disappears, moments latter it reappears on the Canvas, the delay encountered is due (according to the manual and I have no reason to doubt them) the canvas being saved for the Undo function. Pressing the right hand mouse button and squiggling the pointer about causes a trail to be left where the pointer was, this mode the 'freehand thin line drawing' is the the default command upon loading. The command area being at the top of the screen left only the bottom half for drawing. This section I discovered scrolls up and down as you start to push the pointer out of view at the bottom and the top of the screen, screen area permitting of cause. You can also view and work on the whole screen at once by clicking on the 'SHOW' icon, this toggles between the full and partial display. As you move up and down between the command and canvas bounders you jump between the selected screen mode and the command/display screen layout. Experimenting with the other commands soon revels just how powerful the program really is. There are all the usual commands such as circle, ellipse, box, and plot, all those you would expect to find on all art programs. The real power I discovered, came when I started using the brush and spray commands.

Before I go on more about the commands I should explain more about the way the command screen is laid out. From the top down you have the Palette section.This is a band (or bar according to the manual) from which you select the current ink colour. It's divided into 1, 2, 4, or 16 segments depending on what mode your in, the 1 being 256 colour mode. You can only work with the current pallet; they haven't managed to get round that yet, selecting a colour for the palette is a straight forward task just moving the cursor under the colour and clicking, the selected colour then has two lines on top and bottom indicating which is the current colour. Changing the palette colours is just as easy, move the cursor under the colour you want to change and double click (two clicks in quick precession) then voila you are presented with 256 colours displayed in the same way as the 256 colour demo on the Enterprise intro tape. All that is left now is to select a colour by moving a little flashing cursor about till the one you want is found, clicking returns you to the command screen with the new colour in its place. Next bar down holds the stipples, 16 in all, all of which can be changed by the built in stipple editor. Selecting the current stipple is done in the same way as the colours. Beneath the stipples and above the commands sits the brushes, activated and altered in the same way as the stipples.

Going back to the brushes and stipples, you basically just select the brush and the stipple you want to use and draw or brush as the case may be. This may not sound very exciting till you try it, there were so many different combinations of stipples brushes and colour combinations, I was playing with it for over half an hour before I decided to investigated any of the other functions.

Apart some of the more usual functions such as copying and magnify I thought I had mastered all the programs secrets. That was until I looked in the manual. Reading a manuals is for me and I dare say for a lot of other people some thing only done when you've run out of everything else to do. This is were I was pleasantly surprised. Paintbox's manual is surprisingly good, especially when you think of the limited no of sale's you would get from an Enterprise product of this price. Not only does it contain explanations of all the commands and functions (and in a very pleasant readable manor) it covers a whole host of other topics, not just related to using the art program. Amongst hints and tips about cleaning and caring for your mouse you will find a large section explaining how to use the mouse and mouse driver within your own programs, something I will definitely be doing later. As I should of suspected there within the manual were a whole host of extra tricks I had not uncovered in my hours of playing about. Setting markers in a drawing (not visible ones mind) and being able to jump back to them, coupled with being able to lock your X and Y coordinates where just a few listed.

The really exciting tricks did not come from the manual though but from an extra set of instructions, printed on a loose leaf of folded A4 paper. It turns out after the program went on sale, Andrew Richards, undoubtedly the most prolific and arguable the best Enterprise programmer, set to work improving it. The net result is really quite something. He has managed, what to me is a totally new concept in an art programs to enable one drawing to sprayed on to another. That explanation does not make a great deal of sense, try and think about it another way. Imagine you have two pictures, one is painted on a sheet of glass and then laid on top of the other one. Right, you with me so far? You then take a wire brush in the shape of your current brush and scrape away the paint on the top glass sheet, this then revels the one underneath. That, coupled with the ability to copy large chunks of pictures, not necessarily from directly underneath, it is what really makes Paintbox special for me.

The last requirement of a package like this, is producing a hard copy. This request again was handled admirably. With no trouble at all I was churning out reams of paper displaying my master pieces (well maybe not master pieces) in normal, reverse, and in many stages of being squashed (I have yet to find a need for that yet though). I have just realised I've failed to mention one very important ability of Paintbox, you can work in 2, 4, 16, and 256 colour mode, with screens from 2*2 to 42*27 characters in size, . You get a total of 12 screen modes in all. I've get to see that done on another package for any 16 bit micro let alone an 8 bit. I've covered the majority of the functions encapsulated in Paintbox, but still not done it justice. One of thing things that still amazes me still is the flexibility of the package, I'm still finding new ways to create special effects and if I think of something I like produce with a bit of messing around 9 times out of ten, Paintbox will do it.

PROGRAM. Out of This World.

AUTHOR. Andrew Richards.

PUBLISHER. BOXSOFT.

PRICE. £7.95.

Reviewed by Ian Jones.

The idea of the game is to guide your RP32 Transporter Sphere along the roads, avoiding the many pifalls and hazzard which await you. The game is divided into ten levels (each with different hazzards), each having nine roads to be negotiated. When all nine roads of a level are completed (I haven't got this far yet!) you will enter an intermediate game called BREAK OUT. If this is successfully completed you will be rewarded with an extra sphere. Even if it is not completed you will be given a letter (for the first nine levels only). Make a note of these letters and enter them as a command, thus allowing you to enter the tenth and final level. Each level has a name, and these names can also be entered as a Command, thus allowing you to start and play at any point. The first three level names are given (LAND, SEA and AIR), the others have to be discovered as you play. Your sphere can be controlled using the internal or an external Joystick. I have only used it with the internal jpystick, this made the sphere difficult to control due to joystick wobble. This slightly spoilt the game. Not being a natural games player I must admit to finding this game diificult The display within the game is very colourful and the 3D effect of the roads is very good, however, the graphics themselves are quite unimaginative. The first few times I played this game I must admit that I found it rather repetative. However, once you move to the higher levels in the game, you'll be too busy avoiding the hazzards and negotiating the roads to become bored.

Conclusion

I definitely feel that this game is for the hardened games player, its simply too difficult for a beginner like me. The graphics are rather disappointing, but nevertheless are put to good effect. An external joystick is essential for this game as the sphere is very responsive. The ability to enter the level names as commands is a good feature as it allows the rubbishy games players like me to play levels that they would never reach playing the game in sequence. All in all, I would recomend the game to all those Joystick fiends who want something challenging to play.

INTERNATIONAL ENTERPRISE USERS GROUP

Converting Tape-Based Software to Disk

By David Silkstone

Expanding your Enterprise withan EXDOS unit and one or more disk drives is a great experience. No more adjusting the volume to load a tape correctly, no more 'screeching' as you realize that you have left the Tape Sound on, and no more waiting minutes for a game to load. Even copying a game from tape to disk is easy. For those who do not know how to do this follow these steps : (items in single quotes ' should be typed in but the quotes should NOT be included). 1 : Enter ':exdos' in basic to get into the Exdos editor. 2 : Place the disk that you are going to copy to in the drive and change the directory to the one where the program will exist using the CD command (See Exdos manual). 3 : Place the tape that you are going to copy from into the cassette player. 4 : Enter 'copy tape: disk-1:filename' ; where filename is the name that the game will have on the disk. The Enterprise will copy the file from the tape to the disk and give the copied file the name that you specified in the copy command. So far, so good. The problems start when you come to run a program from disk and find the message SEARCHING and the green bar displayed on the status line. YES the program is trying to load the next part from tape (arrrrgh!). What can you do ?. Some programs are quite easy, they have a loader program that is written in basic which has a line that goes something like : RUN "TAPE:" this loads the next part from tape or RUN "TAPE:NEXTBIT" this loads 'NEXTBIT' from tape or OPEN #1:"TAPE:" this opens a channel to tape PRINT USR(LOADER,0) and uses a machine code loader. To alter these programs make a note of the filenames that you gave each part as you copied them, then substitute 'DISK: for TAPE: and insert the correct directory and filename between the colon and the end quotes. e.g. RUN "TAPE:" becomes RUN "DISK:\ADVENTUR\NEXTBIT" Other programs are harder, they use the basic CODE statement to assemble a machine code loader which contains the filename, which has itself been converted into Hexadecimal values. The key to this is to find the sequence '54,41,50,45,3a' which corresponds to 'tape:', and a filename may follow this sequence. (you can use the table on page 23 of issue 7 of Private Enterprise to decode the filename). Substitute the sequence '44,49,53,4b' for the sequence that corresponds to 'tape' as this corresponds to 'disk'.

Remember to save the modified version before trying it, just incase you made a boob somewhere along the line. This is as far as the guidance goes because the more complicated loaders need a program such as 'Devpac' to modify them. Here are some examples of programs that I have converted : 1. Steve Davis Snooker copy the first part from tape as 'SNOOKER' copy the second part from tape as 'SNOOKERA' copy the third part from tape as 'SNOOKERB' in SNOOKER change line 55 to 55 OPEN #8:"DISK:SNOOKERA" line 64 to 64 OPEN $8:"DISK:SNOOKERB" 2. Wordhang copy the first part from tape as 'WORDHANG' copy the second part from tape as 'WORDH1.CHN' in WORDHANG change line 35 to 35 RUN "DISK:WORDH1.CHN" 3. Level 9 Adventures. See the following program. Here is a program called "START" that allows you to select a program from a menu of up to 22 programs. Put the program in the Root directory on disk and just press 'start' to run it. This is the directory of my disk :

```
EXDOS.INI
START \ADV
\ADV\TEXT
FANTASIA
LORDTIME
EMERISLE
FANTASIA.D
SNOWBALL\ADV\GRAPHICS
NODES.ADV
SORCERY
CHARACT.SET
CHARACT.DAT
LOCATION.DAT
GENERAL.DAT
SPRITE.DAT
\ARCADE
\ARCADE\SIMULTN
SNOOKER
SNOOKERA
SNOOKERB
\CHILDS
WORDHANG
WORDH1.CHN
DRAW.BAS
\STRATEGY
CHESS
DECODE.BAS
SQUARES.BAS
```

This is the program that will load the games from a menu :

```
100 PROGRAM "\START"
110 ALLOCATE 1000
120 TEXT 40
130 SET STATUS OFF
140 LET MAX_MENU_SIZE = 22
:! maximum size of menu for 40
column screen
```

```
150 STRING GAME$(MAX_
MENU_SIZE)*18
160 !
170 ! 22 games in menu with
game name a maximum of 18
characters in length
180 !
190 NUMERIC X(MAX MENU_
SIZE),Y(MAX_MENU_SIZE)
200 LET GAMES TOTAL = 12 :!
** CURRENT NUMBER OF
GAMES IN LIST **********
210 LET SPLIT = IP(GAMES
TOTAL/2) :! START OF COLUMN
2
220 LET COLUMN_1 = 1 : LET
COLUMN_2 = 21
230 LET C$ = HEX$("1b,49,2,
8c, c,1b,49,1") :! selection
pointer controls
240 FOR C = 1 TO GAMES_
TOTAL
250   READ GAME$(C)
260   IF C  SPLIT THEN
270     LET X(C) = COLUMN_2 :
LET Y(C) = (2*(C-SPLIT)) + 2
280   ELSE
290     LET X(C) = COLUMN_1 :
LET Y(C) = (2*C) + 2
300   END IF  310   PRINT AT
Y(C),X(C):" " GAME$(C) " ";
320 NEXT C
330 PRINT £102,AT 1,6:C$(1:3)
"Choose game then press
SPACE" C$(6:8);
340 LET DONE = 0
350 LET C = 1
360 PRINT £102,AT Y(C),X(C):
C$(1:4);GAME$(C);C$(5:8);
370 DO
380   DO
390     LET J = JOY(0)
400   LOOP WHILE J = 0
410   PRINT AT Y(C),X(C):" "
GAME$(C) " ";
420   SELECT J
430   CASE 4
440     LET C = C + 1
450     IF C  GAMES_TOTAL
THEN LET C = 1
460   CASE 8
470     LET C = C - 1
480     IF C  1 THEN LET C =
GAMES_TOTAL
490   CASE 1
500     IF C  SPLIT THEN
510       LET C = C + SPLIT
520     ELSE
530       PING
540     END IF
550   CASE 2
560     IF C  SPLIT THEN
570       LET C = C - SPLIT
580     ELSE
590       PING
600     END IF
610   CASE 16
620     LET DONE = 1
630   CASE ELSE
640   END SELECT
650   PRINT £102,AT Y(C),X(C)
:C$(1:4);GAME$(C);C$(5:8);
660 LOOP WHILE DONE = 0
670 SET STATUS ON
680 LET CHOSEN_GAME$ =
GAME$(C)
685 WHEN EXCEPTION USE
LOAD_FAILURE
690   SELECT CASE CHOSEN_
GAME$
700   CASE "Steve D. Snooker"
710     EXT "cd \arcade\simultn"
720     RUN "snooker"
730   CASE "Cyrus Chess"
740     RUN "\strategy\chess"
750   CASE "Nodes of Yesod"
760     RUN "\adv\graphics\
nodes.adv"
770   CASE "Sorcery"
780     EXT "cd \adv\graphics"
790     RUN "sorcery"
800   CASE "Fantasia Diamond"
810     EXT "cd \adv\text"
820     RUN "fantasia"
830   CASE "Emerald Isle"
840     LET FILE$ = "08,45,
4d,45,52,49,53,4c,45"
850     CALL LOAD_LEVEL_9
(FILE$)
860   CASE "Snowball"
870     LET FILE$ = "08,53,4e,
4f,57,42,41,4c,4c"
880     CALL LOAD_LEVEL_
9(FILE$)
890   CASE "Lords of Time"
900     LET FILE$ = "08,4c,4f,52,
44,54,49,4d,45"
910     CALL LOAD_LEVEL_9
(FILE$)
920   CASE "Hangman"
930     EXT "cd \childs"
940     RUN "wordhang"
950   CASE "Mastermind"
960     RUN "\strategy\
decode.bas"
970   CASE "Magic Squares"
980     RUN "\strategy\
squares.bas"
990   CASE "Drawing Program"
1000    RUN "\childs\draw.bas"
6950    ! space here to add extra
case statements
6955    ! dont forget to update
GAMES_total in line  200
6960  CASE ELSE
6965    CLEAR SCREEN
6967    PRINT "Tried to select an
invalid game identifier - check
the case       statements"
6970  END SELECT
6975 END WHEN
6979 !
6980 END
6990 !
7999 !
8000 DATA "Steve D.
Snooker","Cyrus Chess","Nodes
of Yesod","Sorcery","
Hangman","Mastermind","Magic
Squares","Drawing Program"
8010 DATA "Fantasia
Diamond","Emerald
Isle","Snowball","Lords of Time"
8020 !
8999 !
9000 DEF
LOAD_LEVEL_9(FILENAME$)
9010  EXT "cd \adv\text"
9020  CODE = HEX$("4c,6f,61,
64,20,65,72,72,6f,72,2e,20,50,72,6
5,73,73,20,52,45,53,45,54,20,61,6
e,64,20,74,72,79,20,61,67,61,69,
6e,00,00")
9030  CODE = HEX$
(FILENAME$)
9040  CODE LOADER = HEX$
("31,cb,13,3e,fd,d3,b1,21,00,00,22
,f8,bf,01,08,01,16,ff,f7,10,b7,20,2d,
01,0c,01,16,ff,f7,10,b7,20,23,3e,03
,11,f0,12,f7,01,b7,20,19,11,00,18,0
1,00,70,3e,03,f7,06,b7")
9050  CODE = HEX$("20,08,3e,
03,f7,03,b7,ca,00,18,3e,03,f7,03,
3e,66,06,1a,f7,07,3e,66,01,27,00,1
1,c9,12,f7,8,f3,18,fe")
9060  PRINT USR(LOADER,0)
9070 END DEF
9080 !
9090 HANDLER LOAD_FAILURE
9100  CLEAR SCREEN
9110  PRINT "Load Error. press
reset and try again"
9120  CODE LOCK_MACHINE
= HEX$("21,00,00,22,f8,bf,f3,
18,fe")
9130  PRINT USR(LOCK_
MACHINE,0)
9140 END HANDLER NOTE :
```

Sorcery was converted to disk by my brother and the program that does that should appeares opposit.

Fantasia Diamond also loads off a special loader done by my brother which I will get him to send to the user group.

SORCERERS APPRENTICE

With the wave of a joystick you will disk-cover how to cut through all that red tape.

```
100 PROGRAM "sorcery.bas"
101 REM ** You can leave out
all the PRINTS and REMS **
105 CLEAR SCREEN
110 ALLOCATE 9000
120 CODE DUMMY = HEX$("00")
130 LET GAME = DUMMY + 32
140 CODE LOAD = HEX$("3e,
01,11")&WORD$(GAME)&HEX$
("01,93,22,f7,06,c9")
150 CODE DUMP = HEX$("3e,
01,11")&WORD$ (GAME)
&HEX$("01,b0,22,f7,08,c9")
155 PRINT "Put SORCERY tape
into casette and"
156 PRINT "a disk
into the default
drive."
157 PRINT "Press
PLAY on the
tape.":PRINT
160 OPEN
£1:"tape:sorcery"
ACCESS INPUT
165 PRINT
"Loading game
into
memory.":PRINT
170 CALL
USR(LOAD,0)
180 CLOSE £1:LET
COUNT = 0
185 PRINT
"PATCHING
!":PRINT
190 READ ADDR,BYTE
200 DO UNTIL ADDR = 0
210  POKE
GAME + ADDR,BYTE:LET
COUNT = COUNT + BYTE
220  READ ADDR,BYTE
230 LOOP
240 LET CHECK = BYTE
250 IF COUNTCHECK THEN
CALL CHECKSUM:STOP
260 FOR X = 1 TO 7
270  READ SADDR,FADDR
280  LET COUNT = 0
290  FOR ADDR = SADDR TO
FADDR
300    READ BYTE
310    POKE GAME + ADDR,
BYTE
320    LET COUNT = COUNT +
BYTE
330  NEXT ADDR
```

```
340  READ CHECK
350  IF COUNTCHECK THEN
CALL CHECKSUM:STOP
360 NEXT X
370 OPEN £1:"sorcery" ACCESS
OUTPUT
380 CALL USR(DUMP,0)
390 CLOSE £1
399 PRINT "Copying DATA
files":PRINT
400 EXT "copy tape:character_
set charact.set"
410 EXT "copy tape:character_
data charact.dat"
```

```
420 EXT "copy tape:location_
data location.dat"
430 EXT "copy tape:general_
data general.dat"
440 EXT "copy tape:sprite_data
sprite.dat"
450 PRINT "STOP tape":PRINT
460 PRINT "To play SORCERY
just LOAD it.":PRINT
500 END
1000 DEF CHECKSUM
1010  PRINT "Checksum Error :
examine data statements "
1020 END DEF
1497 REM ** Data format **
1498 REM ** address,byte
:repeated **
1499 REM ** stop (0),checksum
**
1500 DATA 2,160,17,131,18,35,
65,32,66,4,71,38,72,4,74,62,76,211
,77,178
```

```
1501 DATA 158,26,236,2,306,
255,360,0,373,0,386,0
1502 DATA 4010,144,4011,16,
4037,144,4038,16,5586,38,5587,4,
5948,144,5949,16
1503 DATA
6930,38,6931,4,6937,21,7411,2,80
78,38,8079,4,8367,11
1504 DATA 8391,46,8395,0,8396,
11,8404,46,8408,0,8409,10,8416,4
6,8420,0
1505 DATA 8630,244,8631,2,
8679,3,8692,21,8705,6,8767,11,0
1506 DATA 2224 ! checksum
1507 REM ** Data  format **
1508 REM ** From,To :
addresses **
1509 REM ** bytes : checksum
**
1510 DATA 287,294,201,0,0,0,0,
0,0,0
1511 DATA 201
1520 DATA 816,873
1521 DATA 205,80,2,195,133,16,
1,2,1,17,1,1,62,7,247,11,195,210,3
4,0,0,62,255
1522 DATA 211,178,62,0,211,
177,62,0,211,179,0,49,255,255,251
,205,20,1,6,10
1523 DATA 118,16,253,33,53,4,
34,248,191,205,39,1,205,11,33
1524 DATA 5524
1530 DATA 3990,4004,185,2,17,
40,0,25,34,144,16,201,0,0,0,0,0
1531 DATA 664
1540 DATA 5916,5918,0,0,0
1541 DATA 0
1550 DATA 8375,8382,46,68,
65,84,0,0,0,12
1551 DATA 275
1560 DATA
8775,8780,46,83,69,84,0,0 1561
DATA 282
1570 DATA 8851,8879
1571 DATA 49,255,63,251,247,
24,32,6,121,50,58,4,247,24,194,73
,35,121,50,62,4
1572 DATA 195,53,4,0,0,0,0,0
1573 DATA 2222
```

HOW TO BUILD YOUR OWN SCART MONITOR CABLE

By David Silkstone

If anyone is thinking of buying a monitor and would like to get one with a SCART socket (like the Enterprise/Fidelity monitor), but is not sure of how to make the cable, here are the instructions (based on the experience I have gained through building my own) :

Items needed :

1 Female card-edge connector with finger pitch of 0.1" (2.5 mm). Make sure you get the type with connectors on both sides and cut it down to 7 connectors in length with a junior hacksaw.

1 to 2 metres of 10-core screened cable, depending on how long you need your cable to be.

6" of good quality insulated wire, or a couple of inches cut off your cable.

1 SCART plug, obtainable from electronic component suppliers such as Maplin, Farnell or Tape Recorder Spares.

1 470 ohm resistor. It should have 4 coloured bands : yellow, purple, brown, gold in that order.

1 pair of extra hands, or failing that a vice to hold the pins of the SCART plug while you crimp and solder them.

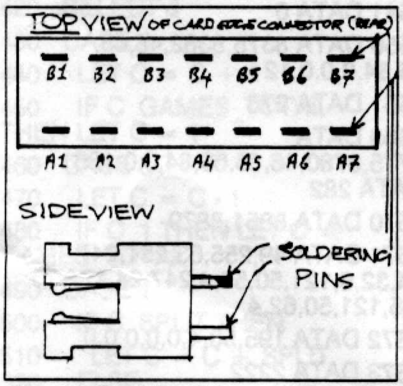


Fig 1. Card edge connector viewed from rear and side.

THE ENTERPRISE END OF THE CABLE (see Figure 1)

It should be quite obvious which side of the connector the cable is attatched to. Decide which is going to be the top of the connector when it is plugged into the Enterprise and mark it so that you don't forget. Place the connector in a vice with the soldering pins upward and the 'top' facing away from you (this is the orientation of Fig. 1). Carefully strip 1" of the outer insulation from the cable, then strip about 0.2" from each of the individual wires inside. Solder nine of the wires, one each, to the following nine pins:

B2, B3, B4, B5, B6, B7, A1, A2, A7.

Make a note of the colour of the wire that you have soldered to each pin.

THE MONITOR END OF THE CABLE

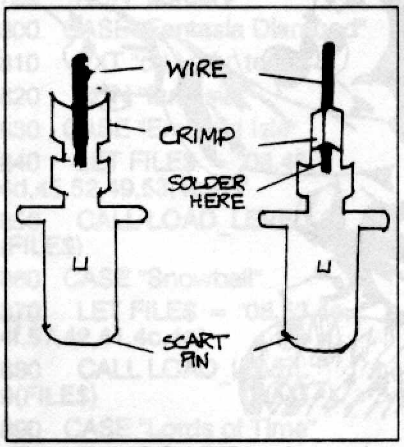


Fig 2. SCART pin wire connection.

Open the SCART plug, slide the threaded collar (small end first) over the cable now, as you will not be able to once you have soldered the wires. Strip about 1.5" of the outer insulation from the cable and about 0.2" from each of the individual wires.

The pins for the SCART plug are loose and lock into position when placed into the body of the plug, so any wiring has to take place before the pins are inserted into the body (See Figure 2 on how to connect the wire and pin together).

Crimp and solder all the single wire connections to the SCART pins :

B3, B4, B5, B7, A1, A7.

Refer to your notes as to which colour wire was connected to which pin of the card-edge connector. Next, get two pieces of wire of about 0.5" in length and solder one to each end of the resistor. Twist and solder the end of one of these wires with the wire from B6, then crimp and solder the ends of to the SCART pins (see Figure 3).

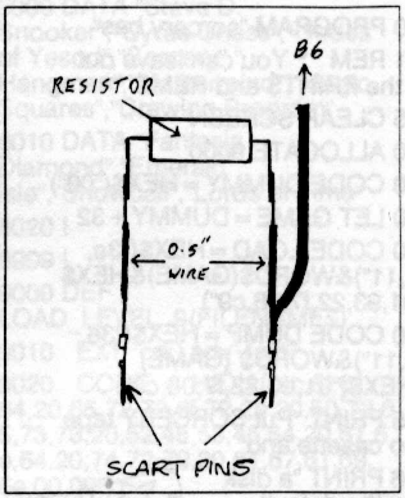


Fig 3. Resistor to SCART pin detail.

Prepare a set of linked wire pieces, each about 0.75" long (see Figure 4).

To this set solder the wire from B2 to one end of the set, then solder the wire from A2 to the second link in, starting from the other end (see Figure 5) . To each link in this chain then crimp and solder a SCART pin (ie. 5 pins).

Insert the remaining unused SCART pins into the body of the plug in the slots that are not going to be used (see Figure 6). Use a small tool to push the pins so that they latch into place.

Using the same technique and following Figure 6, insert the wired pins into the body of the SCART plug, starting with the resistor pair of pins, then the 5-pin chain and finally the individually wired pins.

Snap the cover of the plug back over the body and thread the collar back onto the cover. Connect the cable to your monitor and Enterprise (remember, you marked the top of the card edge connector, didn't you ?). If you have followed the instructions carefully you should now have a working monitor cable !

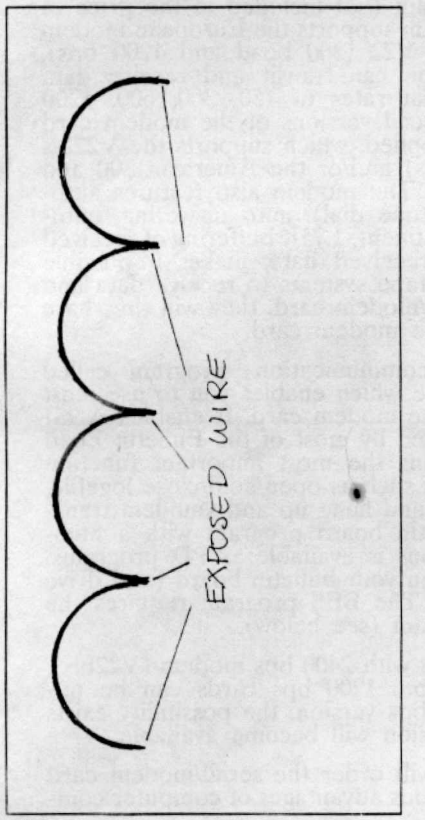


Fig 4.

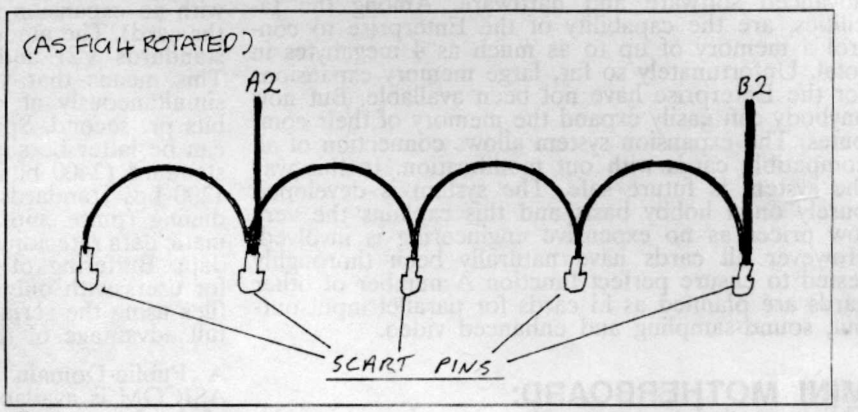


Fig 5.

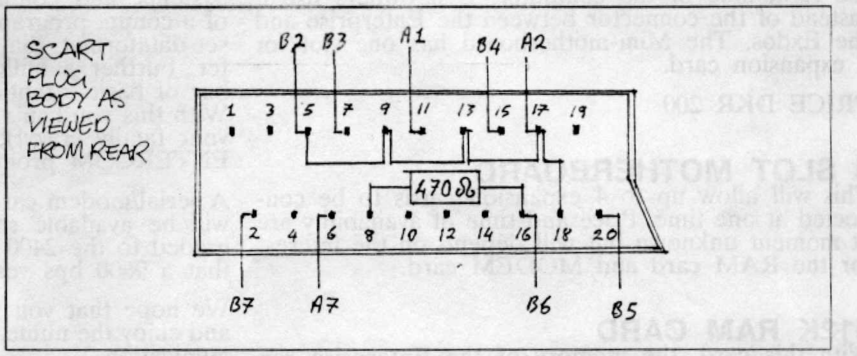


Fig 6. SCART plug body veiwed from rear.

RESUME OF CABLE CONNECTIONS

SCART PLUG	FUNCTION	EXTRA CONNECTIONS	ENTERPRISE
2	Right hand audio IN		B7
5	Blue signal earth	(+9,13,17,18)	B2
6	Left hand audio IN		A7
7	Blue signal IN		B3
8	Status CVBS	(resistor to 16)	
9	Green signal earth	(+5,13,17,18)	
11	Green signal IN		A1
13	Red signal earth	(+5,9,17,18)	
15	Red signal IN		B4
16	Status RGB	(resistor to 8)	B6
17	CVBS earth	(+5,9,13,18)	A2
18	RGB status earth	(+5,9,13,17)	
20	CVBS earth		B5

BUG TRONICS, in cooperation with DATAQUIP ELECTRONICS England has developed hardware expansions for the Enterprise, which will enhance the use of your machine. As you know, the Enterprise isn't any old home computer, but a computer with both advanced software and hardware. Among the facilities, are the capability of the Enterprise to control a memory of up to as much as 4 megabytes in total. Unfortunately so far, large memory expansions for the Enterprise have not been available. But now anybody can easily expand the memory of their computes. The expansion system allows connection of all compatible cards with out modification. In this way the system is future safe. The system is developed purely on a hobby basis and this explains the very low prices as no expensive engineering is involved. However all cards have naturally been thoroughly tested to ensure perfect function A number of other cards are planned as f.i cards for parallel input output, sound-sampling and enhanced video.

MINI MOTHERBOARD:

This is needed to connect the external cards to the Enterprise and is connected to the expansion bus on the right side of the computer. Disk-owners use it instead of the connector between the Enterprise and the Exdos. The Mini-motherboard has one slot for a expansion card.

PRICE DKR 200

4 SLOT MOTHERBOARD.

This will allow up to 4 expansion cards to be connected at one time. Price and time of availability are at moment unknown but will depend on the interest for the RAM card and MODEM card.

512K RAM CARD

With this card, the memory of the Enterprise can be expanded with 512K. Several cards can be connected to give as much as possible, up to 2 megabytes totally. You may often have been irritated of the message "Insufficient memory", which you will hardly see again. Owners of Enterprise 64's will gain a 20-25% speed increase, and fortunate owners of the disk system will be able to make huge Ramdisks. (max. 2000k using four cards) A huge ramdisk is very useful with programs accessing disks frequently. Using the speed and save your disk-drives. You will also now have space for a lot of extension-programs, large printer buffers, superfast graphic dumps, and disk operations like "copy" will run much faster. A large memory will also allow many basic programs to reside in memory at that same time. 512K f.i. gives space for about 30 programs!

PRICE ONLY DKR 1150 (not including mini motherboard).

DEDICATED SERIAL/MODEM CARD

For a long time Enterprise users have been excluded from the field of online communication using the public telephone net. Even though the Enterprise has a serial port, it cannot receive and transmit simultaneously (full duplex), which is necessary when communicating through a modem. This problem has now been solved by the serial/modem card, which contains not only an ON-CARD PROFESSIONAL MODEM but also a standard full-duplex serial port.

The serial/modem card is connected externally to the Enterprise bus and requires only a mini motherboard with an expansion slot (not included in the price of the card). The modem supports the European modem standards V21 and V22 (300 boad and 1200 bps). This means that you can transit and receive data simultaneously at dat rates of 150, 300, 600, 1200 bits pr. second. Special versions of the modem card can be latter be supplied, which supports the V22bis standard (2400 bits/s) and/or the American 300 and 1200 bps standards. The modem also features autodialing (pulse and tone dial), auto-answering, automatic data rate adjustment, 1.75K buffering of received data. Buffering of received data, makes it possible for users with only tape systems to receive data and files using the serial/modem card. they will thus have full advantage of the modem card.

A Public-Domain communications program called ASICOM is available which enables you to use most of the features of the modem card. It enables ANSI escape-sequences used by most of the Bulletin Boad Systems, and contains the most important function of a comm. program such as open/hold/close logefile, set dataformat, dial and hang up and xmodem transfer. Further a bulletin board program with a number of basic-extentions is available as PD programs. With this you can run your bulletin board (and drive your family crazy!). The BBS program requires the ENTERCOM program (see below).

A serial/modem card with 2400 bps modem (V22bis), will be available soon. 1200 bps cards can be upgraded to the 2400 bps version. the possibility exists that a 9600 bps version will become available.

We hope that you will order the serial/modem card and enjoy the numerous advantages of computer communication.

Prices: Standard SERIAL/MODEM CARD: Dkr 1400
Estimated extra price for V22bis version/upgrade is Dkr. 400.-

ENTERCOM

Entercom is a program which is used in conjunction with the serial-modem card described above. It contains two new drivers; one for the modem and one for the serial port on the card. It also contains an advanced communications program which can handle both text and graphics. The device drivers will allow you to use all of the features of the serial/modem card in a natural way. You can f.i. Load programs from MODEM:, enabling you to directly (down) load programs from the telephone line just as if you were loading it from DISK:/TAPE:, copy from COM: (the RS232 serial port) to file etc.

Price for ENTERCOM program is DKR 250.-

Orders and further
information from:

BUG TRONICS F.& P. Stadel Nielsen,
Skovskellet 35 A,
DK-2840 HOLTE DENMARK
TEL 45-2-800745
Postal account (Postscheckkonto, postgiro) 9 85 78 26

All deliveries only by prepayment or Cash on delivery (Nachahme, postpkravning), (postage extra Dkr 25.-) Cheques in foreign currency will not be accepted! Do not send money with out conformation from us. As our cards are totally handmade, a delivery time of 2-3 weeks must be expected. We reserve all rights to change specifications and prices with out notice.